

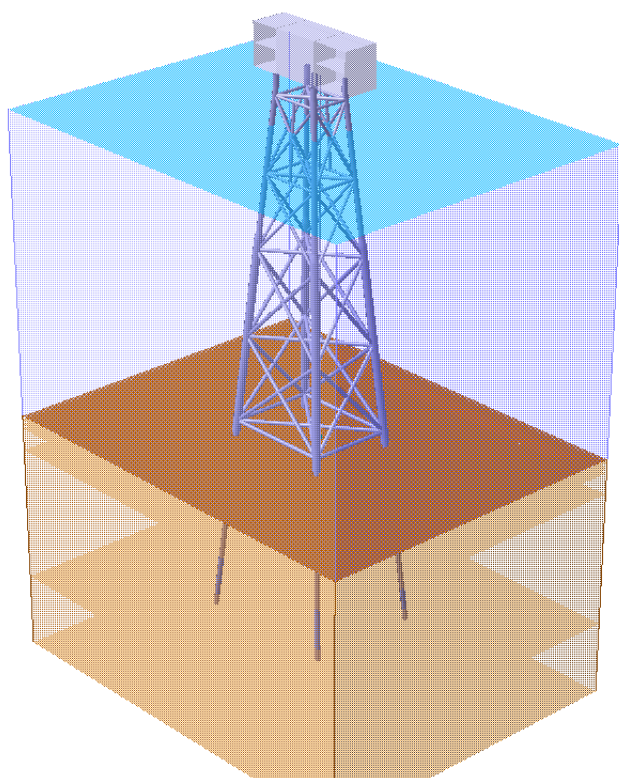


---

# SESAM USER MANUAL

---

## GeniE Vol. 2



GeniE for Wave and Structure-Pile-Soil Analysis

DET NORSKE VERITAS



SESAM  
User Manual  
**GeniE Vol. 2**

GeniE for Wave and Structure-Pile-Soil Analysis

September 15th, 2005

Valid from program version 3.2

Developed and marketed by  
DET NORSKE VERITAS

## DNV Software Report No.: YY-NNNN / Revision 1, September 15th, 2005

Copyright © 2005 Det Norske Veritas

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Published by:

Det Norske Veritas  
Veritasveien 1  
N-1322 Høvik  
Norway

Telephone:	+47 67 57 99 00
Facsimile:	+47 67 57 72 72
E-mail, sales:	<a href="mailto:software.sesam@dnv.com">software.sesam@dnv.com</a>
E-mail, support:	<a href="mailto:software.support@dnv.com">software.support@dnv.com</a>
Website:	<a href="http://www.dnvsoftware.com">www.dnvsoftware.com</a>

If any person suffers loss or damage which is proved to have been caused by any negligent act or omission of Det Norske Veritas, then Det Norske Veritas shall pay compensation to such person for his proved direct loss or damage. However, the compensation shall not exceed an amount equal to ten times the fee charged for the service in question, provided that the maximum compensation shall never exceed USD 2 millions. In this provision "Det Norske Veritas" shall mean the Foundation Det Norske Veritas as well as all its subsidiaries, directors, officers, employees, agents and any other acting on behalf of Det Norske Veritas.

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1-1</b>
1.1	GeniE Vol. 2: Wave and Structure-Pile-Soil Analysis .....	1-1
1.2	How to read the Manual.....	1-1
1.2.1	Terminology .....	1-2
1.3	Release Notes and Status List.....	1-2
<b>2</b>	<b>FEATURES OF GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS.....</b>	<b>2-1</b>
2.1	Basic Features .....	2-1
2.2	Wind, Wave and Current Modelling and Analysis.....	2-2
2.3	Piled Foundation Modelling and Analysis.....	2-2
<b>3</b>	<b>USER'S GUIDE TO GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS</b>	<b>3-1</b>
3.1	Introduction to Wind, Wave and Current Load Analysis .....	3-2
3.2	Getting Started doing Design Wave Analysis .....	3-2
3.2.1	Environment Data.....	3-3
3.2.2	Hydro Properties.....	3-8
3.2.3	Define the Design Wave Analysis.....	3-11
3.2.4	Load Case names, Description, FEM Loadcase numbers and FEM LC Rule .....	3-15
3.2.5	Execution of the Design Wave Analysis.....	3-17
3.2.6	Execution of the Design Wave Analysis with Modified Input .....	3-20
3.3	Getting Started doing Deterministic Fatigue Wave Analysis .....	3-21
3.3.1	Environment Data.....	3-21
3.3.2	Hydro Properties.....	3-24
3.3.3	Define the Deterministic Fatigue Wave Analysis .....	3-25
3.3.4	Execution of Deterministic Fatigue Wave Analysis .....	3-27
3.4	Getting Started doing Stochastic (Spectral) Fatigue Wave Analysis .....	3-28
3.5	Introduction to Structure-Pile-Soil Analysis.....	3-28
3.5.1	Concepts in a Structure-Pile-Soil Analysis .....	3-28
3.5.2	Modelling .....	3-30

3.5.3	Analysis .....	3-33
3.5.4	Results .....	3-33
3.6	Getting Started doing Structure-Pile-Soil Analysis with Design Wave Loads.....	3-34
3.6.1	Environment .....	3-34
3.6.2	Scour.....	3-35
3.6.3	Soil Properties .....	3-35
3.6.4	Pile Modelling .....	3-39
3.6.5	Define the Pile-Soil Analysis .....	3-43
3.6.6	Executing the Pile-Soil Analysis .....	3-45
3.6.7	Reviewing Analysis Results .....	3-46
3.6.8	Execution of the Pile-Soil Analysis with Modified Input .....	3-47
3.7	Doing the Analysis Outside GeniE .....	3-48
<b>4</b>	<b>EXECUTION OF GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS...</b>	<b>4-1</b>
4.1	Program Environment.....	4-1
4.2	Files used by GeniE .....	4-1
<b>5</b>	<b>DIALOG DESCRIPTION .....</b>	<b>5-1</b>
5.1	Dialog Features .....	5-1
5.1.1	Dialog Help .....	5-1
5.1.2	Dialog New/Edit Functionality.....	5-2
5.1.3	Late Definition Functionality .....	5-2
5.1.4	Table Functionality.....	5-3
5.2	The Environment .....	5-5
5.2.1	Location.....	5-5
5.2.2	Directions .....	5-7
5.2.3	Air.....	5-8
5.2.4	Water .....	5-9
5.2.5	Soil.....	5-19
5.3	The Hydro Properties.....	5-24
5.4	The Wave Load Analysis.....	5-39
5.4.1	The Analysis Activity.....	5-39
5.4.2	The Wave Load Run.....	5-40
5.5	Execution of a Wave Load Analysis.....	5-49
5.6	Execution of a Structure-Pile-Soil Analysis .....	5-51
<b>APPENDIX A</b>	<b>TUTORIAL EXAMPLES.....</b>	<b>A-1</b>
A 1	Fixed 4-Legged Jacket Input for Design Wave Analysis .....	A-1
A 2	Fixed 4-Legged Jacket Input for Deterministic Fatigue Wave Analysis.....	A-12

A 3	4-Legged Jacket Input for Structure-Pile-Soil Analysis.....	A-24
-----	---	------





# 1 INTRODUCTION

## 1.1 GeniE Vol. 2: Wave and Structure-Pile-Soil Analysis

This is the user manual for the part of GeniE dedicated to wind, wave and current load analysis as well as structure-pile-soil analysis of offshore jackets.

This user manual assumes that the user has knowledge in the use of GeniE as covered by the GeniE User Manual Vol. 1 (the main user manual of GeniE). Furthermore, knowledge in the use of Wajac for wave load analysis and Sestra and Splice for structure-pile-soil interaction analysis is also required.

The manual covers two main areas of use of which the latter is normally in combination with the former:

- Wind, wave and current load analysis: modelling for and execution of Wajac
- Structure-pile-soil analysis: modelling for and execution of a Sestra-Splice interaction analysis

Wind, wave and current load and structure-pile-soil analysis in GeniE is performed by defining the model at a conceptual level and then letting the program derive an analysis representation. The analysis representation will consist of a FE model (beam elements and optionally plate elements for the topside structure), wave load conditions, piles, soil data, etc.

## 1.2 How to read the Manual

If you are a new user then read Chapter 2 to learn about basic features and principles of GeniE for wind, wave and current load and structure-pile-soil analysis.

If you already have a basic understanding of GeniE for wind, wave and current load and structure-pile-soil analysis then go to the appropriate ‘Getting Started doing ...’ section of Chapter 3:

- Section 3.2 Getting Started doing Design Wave Analysis
- Section 3.3 Getting Started doing Deterministic Fatigue Wave Analysis

- Section 3.4 Getting Started doing Stochastic (Spectral) Fatigue Wave Analysis (not available in GeniE version 3.2)
- Section 3.6 Getting Started doing Structure-Pile-Soil Analysis with Design Wave Loads

Chapter 4 EXECUTION OF GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS explains which files are used by GeniE and possible limitations.

Chapter 5 DIALOG DESCRIPTION explains in detail all dialogs of GeniE for wind, wave and current load and structure-pile-soil analysis.

### **1.2.1 Terminology**

This document uses some terms, acronyms and abbreviations that are explained below.

- SWL = Still water level
- Seabed = Seafloor, mudline
- Hydro property data = Hydrostatic and hydrodynamic property data

## **1.3 Release Notes and Status List**

There exists for GeniE (as for all other SESAM programs) information additional to this user manual. This may be:

- Reasons for update (new version)
- New features
- Errors found and corrected (bug fixes)
- Etc.

The main source for information on new features and bug fixes for GeniE is the Release Notes available through Help > Help Topics (F1) in GeniE.

Additionally, there is a Status List for GeniE found on our website [www.dnvsoftware.com](http://www.dnvsoftware.com). Go to services > support and click the SESAM Status Lists link and log into this service. Contact us for log-in information.

## **2 FEATURES OF GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS**

### **2.1 Basic Features**

GeniE for wind, wave and current load and structure-pile-soil analysis allows you to model and describe the environment (water and air) as well as the piled foundation of an offshore jacket structure. This includes:

- Wind modelling
  - Wind speed
  - Air drag coefficients
- Hydrodynamic modelling
  - Waves
  - Current
  - Hydrodynamic (Morison) coefficients
  - Marine growth
  - Flooding
  - Etc.
- Piled foundation modelling
  - Piles
  - Soil

GeniE also controls the execution of Wajac (wind, wave and current load analysis), Sestra (linear structural analysis) and Splice (non-linear pile-soil analysis).

## **2.2 Wind, Wave and Current Modelling and Analysis**

Three types of wave load analysis are available:

- Design wave
- Deterministic time domain
- Frequency domain

**Note:** Frequency domain analysis is disabled in the 3.2 version of GeniE. Frequency as well as irregular time domain analyses are subject to implementation in later versions of GeniE.

## **2.3 Piled Foundation Modelling and Analysis**

The soil with sand and clay layers may be modelled together with the piles. GeniE supports the execution of all programs involved in an integrated structure-pile-soil analysis.

### **3 USER'S GUIDE TO GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS**

This chapter describes how to use GeniE for:

- Wind, wave and current load analysis
- Structure-pile-soil analysis

For both of these two main areas of use there is an introductory section, Section 3.1 and Section 3.5, respectively. The introductory sections are followed by sections of type 'Getting Started ...' that explain by way of examples how to use GeniE for the various tasks.

### 3.1 Introduction to Wind, Wave and Current Load Analysis

A wind, wave and current load analysis requires different types of data to be entered. These are environment data, 'hydro' (hydrostatic and hydrodynamic) property data and run data. In the following the basic input is explained for three different types of analysis. All dialogs and commands are described in detail in Chapter 5 DIALOG DESCRIPTION.

The following three sections describe by way of examples one type of analysis each.

- Section 3.2 Getting Started doing Design Wave Analysis
- Section 3.3 Getting Started doing Deterministic Fatigue Wave Analysis
- Section 3.4 Getting Started doing Stochastic (Spectral) Fatigue Wave Analysis

The examples show that sets are useful when assigning hydro properties to the structure. Hence, you will benefit by initially creating named sets corresponding to the various sets of properties. For example, if the structure is to be assigned different Constant Marine Growth properties in different depth intervals then create a set for each depth interval.

All analysis types make use of the same jacket model including a topside structure with Z-coordinate ranging from 0 m to 160 m. The seabed (mudline) is at 0 m and the sea surface at 124 m.

The wave load analysis will be performed by the hydrodynamic analysis program Wajac. Wajac calculates wind, wave and current loads on fixed and rigid frame structures. The frame structure model is transferred to Wajac through the Input Interface File (FEM-file). The hydro properties Constant Morison coefficients and Flooding are also transferred via this file. All other input to Wajac comes from the Wajac.inp file created by GeniE. The Wajac.inp file is open for modification by the user though. See Section 4.2 about files used by GeniE.

### 3.2 Getting Started doing Design Wave Analysis

This is a description of how to do a design wave analysis by way of a simplified example. The input of environment, hydro property and wave load run data is described. All other necessary data is assumed to exist, i.e. jacket structural data with properties and loads other than hydro properties and wave loads.

If you want to perform the operations described below then go to Section A 1 for the input of the structural data. (The input may be copied from an electronic version of the manual.)

This design wave analysis consists of five seastates: four waves including current and wind plus a buoyancy calculation. Current and wind are in the same direction as the wave.

Wave load data is defined in the order:

- Environment data – Section 3.2.1
- Hydro properties – Section 3.2.2
- Analysis activity – Section 3.2.3

### 3.2.1 Environment Data

These data describe the environmental part of a wave load analysis, e.g. wind, current, waves and water depth in addition to constants like gravity and water density. The following environment data are defined:


- Location
- Current profile
- Wind profile
- Regular wave set
- Wave load condition

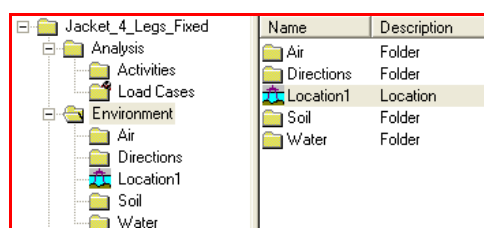
The wave load condition defines how the environment data will be used in the analysis.

#### Define the Location

It is natural to start by defining the location. Right-click (click right mouse button on) the Environment folder and select New Location (or Insert > Environment > Location). The Location dialog (Figure 5.6) appears with modifiable default values. Let the values in the Air tab be as is. In the Water tab set Waterline Z to 124 m and in the Soil tab set Seabed Z (mudline) to 0 m.

**Note:** The Waterline Z and Seabed Z values are given in the model (global) coordinate system.

Click OK to create the Location and see a Location icon appearing in the Environment folder as shown in Figure 3.1. Also, the seabed and SWL (still water level or waterline) appear in the display as shown in Figure 3.2. Colours, transparencies and whether an object is visible or not are easily modified by right-clicking the Location selection button  in the toolbar and selecting the appropriate item. If, for example, the air layer is visible it will normally extend far above the structure and you may want to remove it. Right-click the Location selection button to turn on/off the visibility of air. Clicking (with the left mouse button) the Location selection button enables (when depressed) and disables (when lifted) selection of environment items in the same way as the other selection buttons.



**Figure 3.1** The location icon in the Environment folder

22 Nov 2004 14:04  
Jacket\_4\_Legs\_Fixed  
WindW

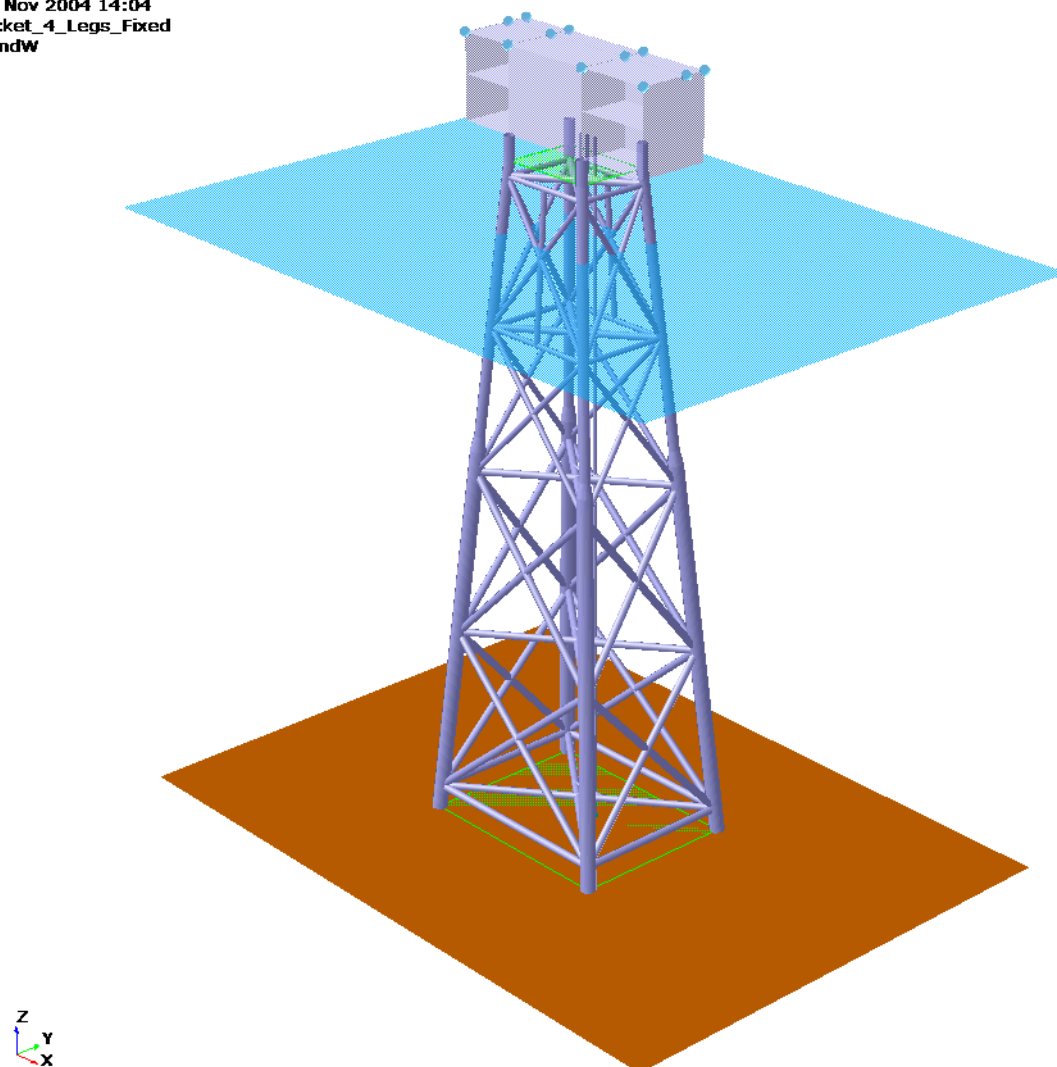


Figure 3.2 The location displayed with water surface and seabed (water made invisible)



## Define the Current Profile

Define a current profile by right-clicking the Environment > Water folder and selecting New Current Profile; see Figure 3.3. Note that information on how to enter data is provided by positioning the cursor over the light bulb icon; see Figure 3.4. In the Create/Edit Current Profile dialog the Direction along wave heading option is used, therefore, only one current profile is needed. This option disables the Direction column. The current must be defined for the wet part, i.e. above maximum wave height which in this case is 28 m. The elevations must be given in sequence of decreasing Z-values and the Z-value is always 0 at sea surface (SWL) and decreasing downwards. The current profile to enter is as shown in Figure 3.3. Click Apply and then Cancel and find the Current Profile in the Environment > Water folder.

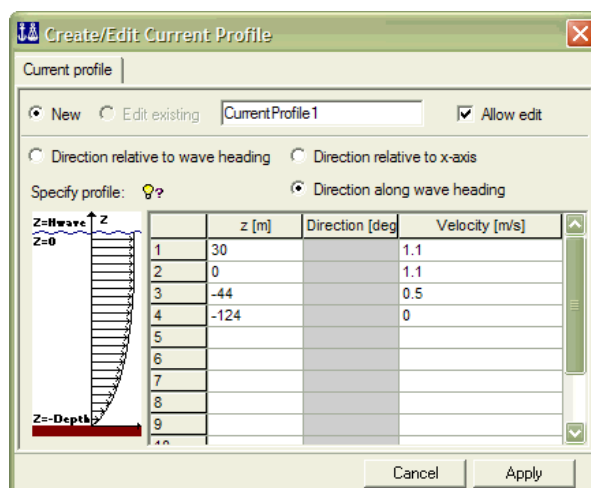


Figure 3.3 The Create/Edit Current Profile dialog

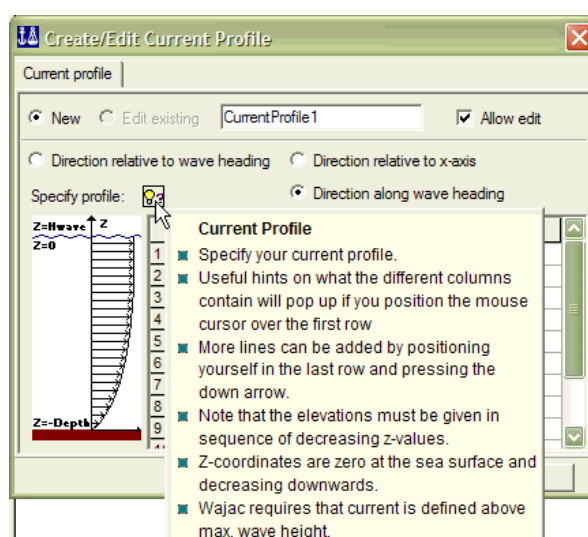


Figure 3.4 The cursor over the light bulb icon provides information on how to enter a current profile

## Define the Waves

A Regular Wave Set constituted by wave components is defined. Each component consists of period, height, phase angle and direction data. The phase angles will serve as initial phase angles in the seastates defined in the Edit Wave Load Run dialog (Figure 5.41).

Right-click the Environment > Water folder and select New Regular Wave Set; see Figure 3.5. Note that general information on the dialog is provided by positioning the cursor over the book icon as shown to the right in Figure 3.5. Select the Wave Set Type Period radio button. The Fill tools are for filling in data for Period, Height, Phase and Direction. See the description accompanying Figure 5.15 for how to use this tool. In this example there are only four wave components so the data are entered directly into the table as shown in Figure 3.5.

Click OK and see that the Regular Wave Set appears in the Environment > Water folder.

**Note:** Defining a Regular Wave Set does not involve creation of any load cases, i.e. no wave loads will appear in the Analysis > Load Cases folder. Wave load cases are not created until defining or editing the wave load run, see Section 3.2.3.

Regular Wave Set

Name: WaveSet1

Wave Set Type: ☒ Period ☐ Wavelength ☐ Frequency ?

Fill tools: ☒ Period ☐ Height ☐ Phase ☐ Direction ?

Sequence:

First value: [ ] [s] ✖

Last value: [ ] [s] ✖

Step value: [ ] [s] ✖

Fill table ☐ Combine all with all ?

Single value:

Value: [ ] [s] ✖

Fill all ?

Fill selected ?

Fill equal components ?

Specify value: ?

	Period [s]	Height [m]	Phase [deg]	Direction [deg]
1	14	26	-60	270
2	12	15	-60	180
3	14	28	-60	90
4	15	27	-60	0
5				
6				
7				
8				
9				
10				

Clear table Remove Insert OK Cancel Apply

Regular Wave Set

Name: WaveSet1

Regular Wave Set dialog

- With the regular wave set dialog the user defines an arbitrary number of wave components.
- Each wave must contain values for period, height, phase and direction.
- The wave components may either be edited manually into the table, or the "Fill tools" may be used.

General information provided

**Figure 3.5** The Regular Wave Set dialog, general information provided by the book icon

Define the Wave Load Condition

The Wave Load Condition defines how the basic environmental data will be used in the analysis. A Condition of type Deterministic Time is created. Right-click the Environment > Location1 icon and select New Wave Load Condition; see Figure 3.6. The Deterministic Time radio button is currently the only option.

First tick the Regular wave set check box and select the previously defined WaveSet1 from the combobox in the Wave components area. The wave components appear in the table. In this example we have only four components and will, therefore, not use the Assign wave component properties tool (this tool will be used in Section 3.3 Getting Started doing Deterministic Fatigue Wave Analysis). For each wave component select current profile (the only one defined), wind profile (none in this example) and wave model (Stokes5, i.e. Stokes 5th order wave theory) by combobox menus in the table (when clicking a cell a combobox appears). You may need to stretch the dialog somewhat. Add a fifth component containing no wave data and with wave model CalmSea. This last row of the table will be a separate buoyancy calculation. Click OK and see that the Wave Load Condition (Condition1) appears in the Environment > Location1 folder.

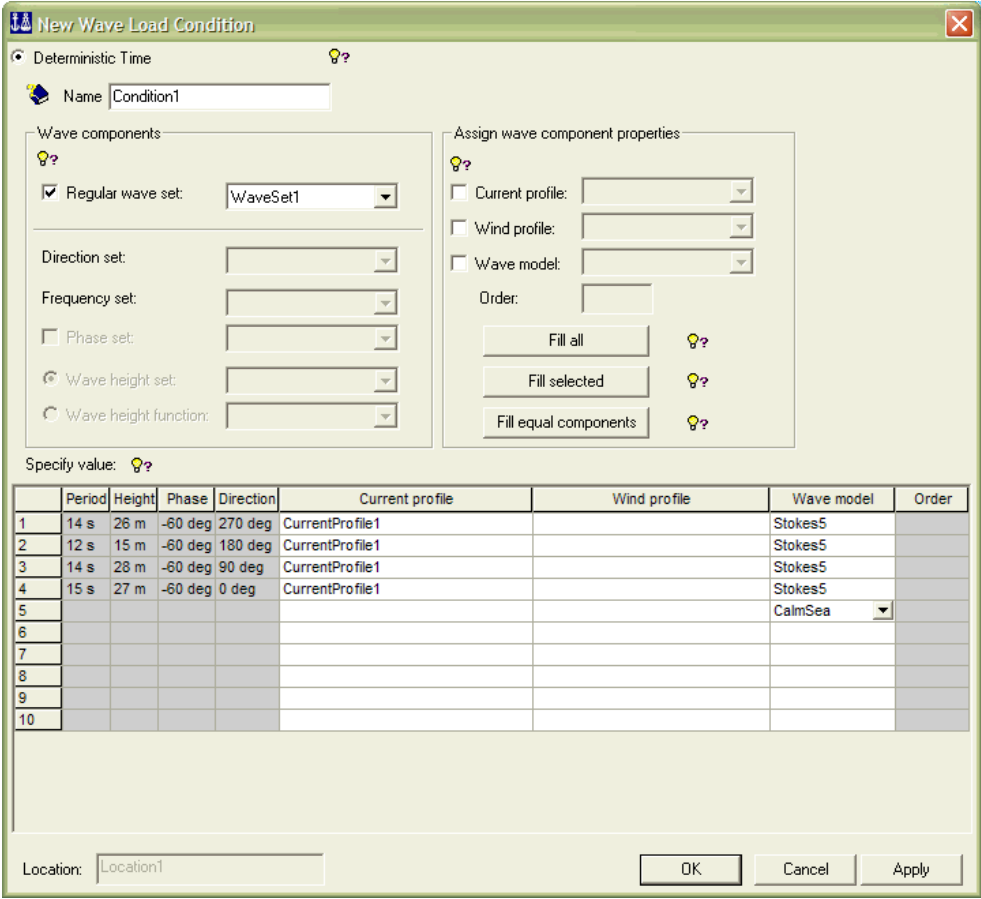
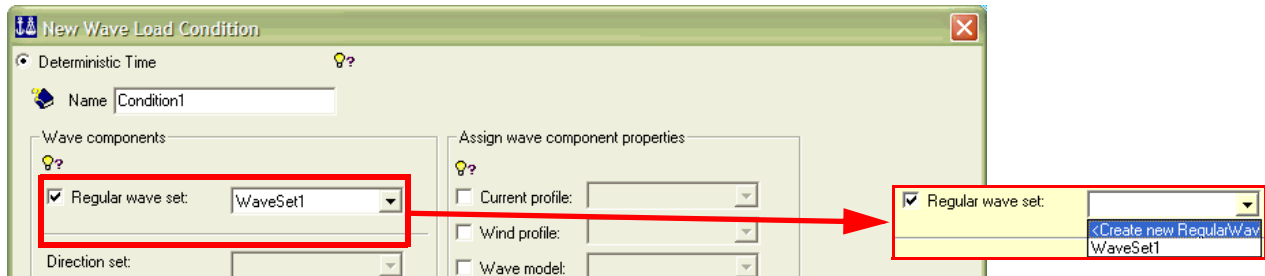


Figure 3.6 The New Wave Load Condition dialog

Dialogs for defining waveset, current and wind profile objects may be activated directly from the New Wave Load Condition dialog. For example, the Regular Wave Set dialog is activated from the Regular wave set combobox in the Wave components area by selecting <Create new RegularWaveset>. See Figure 3.7.

Thus, no data need to be defined up front. This may be the preferred way to work when you are familiar with the functionality. Several dialogs offer such functionality.



**Figure 3.7 Go to Regular Wave Set dialog from the New Wave Load Condition dialog**

**Note:** Defining the Wave Load Condition does not involve creation of any load cases, i.e. no wave loads will appear in the Analysis > Load Cases folder. Wave load cases are not created until defining or editing the wave load run, see Section 3.2.3.

### 3.2.2 Hydro Properties

Several types of hydro properties may be defined but only two are mandatory, i.e. assuming that both wind and wave loads are to be computed:

- Morison coefficients for the wet part, i.e. up to maximum wave height
- Air drag coefficients for the dry part, i.e. down to still water level

In this example we do not have any wind profile so air drag coefficients are not required.

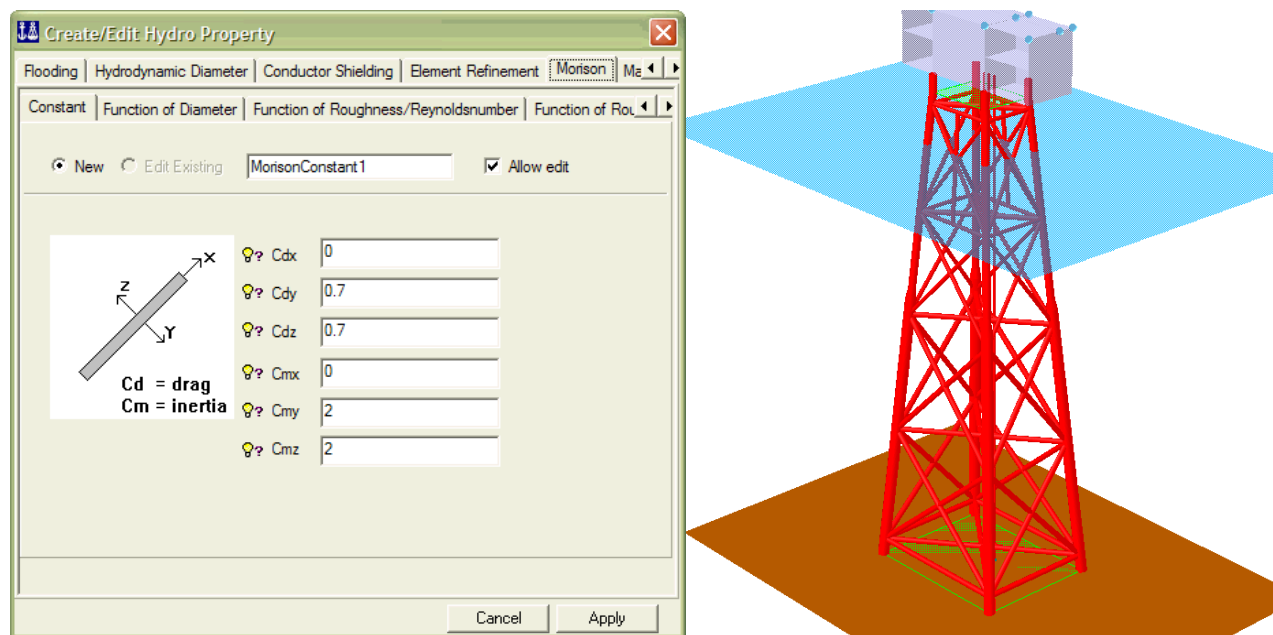
When a property is defined it appears in the Properties > Hydro folder. It has to be connected to the structure. The hydro properties that will be specified in this example are Morison Coefficients, Marine Growth and Flooding. A more detailed description of the hydro properties is found in Section 5.3.

## Define and assign Morison Coefficients

A Morison Constant property is defined and assigned to the structure. In this example only one property using the default values is defined. For simplicity it is assigned to the complete structure even though it is relevant only on the wet part.

Right-click the Properties > Hydro folder and select Morison > New Morison Constant to open the Create/Edit Hydro Property dialog; see Figure 3.8. (You can also access this feature through Edit > Properties > Hydro Property tab > Create/Edit Hydro Property > Morison tab.) The Constant tab appears with default values. Click Apply (and Cancel) and the hydro property MorisonConstant1 appears in the browser.

The property MorisonConstant1 is then assigned to the structure. As previously mentioned it is convenient to have named sets comprising different parts of the structure for the purpose of assigning properties. Select previously defined sets in the Utilities > Sets folder. Alternatively, select the appropriate part of the structure by clicking/rubberband in the display area or by selecting in the browser. The selected part of the structure is highlighted; see Figure 3.8. In the Properties > Hydro folder right-click the MorisonConstant1 object and select Apply Morison coefficients to selection. Morison coefficients are now assigned to the structure.



**Figure 3.8** The Create/Edit Hydro Property > Morison dialog – assign coefficients to the structure

**Note:** If no Morison Coefficient property is assigned, the wave load analysis will fail.

## Define and assign Marine Growth

Marine Growth as Function of Z level is defined and assigned to the structure.

Right-click the Properties > Hydro folder and select New Marine Growth Z-level. (You may also do this by Edit > Properties > Hydro Property tab > Create/Edit Hydro Property > Marine Growth > Function of Z-level tab.) The Function of Z-level tab appears; see Figure 3.9. You may need to stretch the dialog somewhat to see the graph at the right properly. Tick Use marine growth when calculating inertia force as we want the marine growth to contribute to not only the drag force but also the inertia force. After entering the values shown in the figure notice that the graph may show all and any of the three parameters Thickness, Roughness height and Density factor.

**Note:** Z-levels should be given in sequence of decreasing Z-values. Z-value at the still water level is always 0 and Z-axis is pointing upwards.

Click Apply and Cancel.

Conclude by assigning the property to the whole jacket structure shown in Figure 3.8 in the same way as described for the Morison coefficient above.

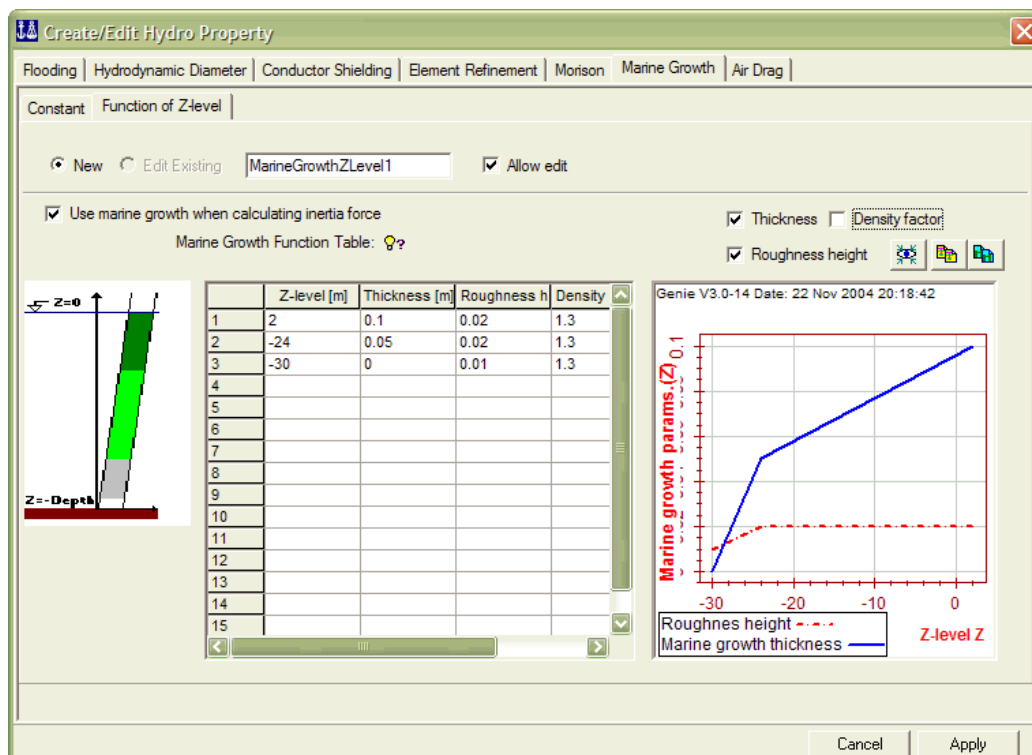


Figure 3.9 The Create/Edit Hydro Property > Marine Growth dialog

## Define and assign Flooding

A Flooding property with Flooding value = 1, i.e. member is flooded, is defined and assigned to the part of the structure being flooded tubes.

Right-click the Properties > Hydro folder and select New Flooding. The Flooding tab appears in which the value 1 is entered; see Figure 3.10.

Click Apply and Cancel.

Only the legs are flooded which means that the Flooding property should be assigned to the legs only. If a set containing the legs exists use this to assign the property in the same way as described for the Morison coefficient above. The View Options dialog (Alt+O) > Model tab allows colour coding the Flooding property for verification purposes; see Figure 3.10.

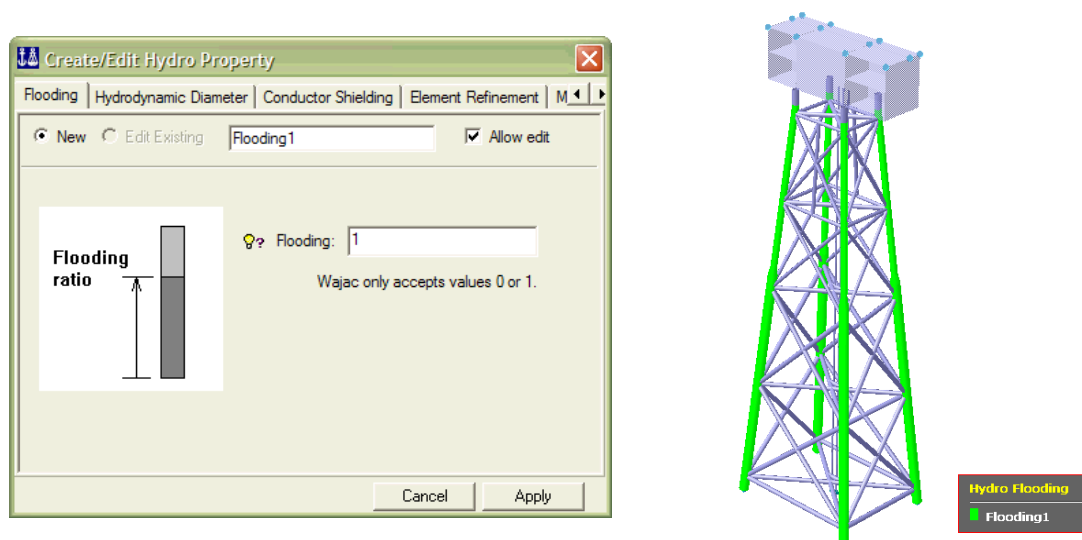


Figure 3.10 The Create/Edit Hydro Property > Flooding dialog

**Note:** In Wajac only completely flooded (value 1) or non-flooded (value 0) is possible. Wajac selects the nearest integer to the value specified, i.e. 1 or 0.

**Note:** By default all beams are non-flooded in Wajac, i.e. only flooded beams need to be specified.

### 3.2.3 Define the Design Wave Analysis

To perform a Design Wave Analysis an activity must be created. Right-click the Analysis > Activities folder and select New Analysis (or Alt+D). In the dialog appearing the Linear Structural Analysis option is already selected. Tick the Wave Load Activity. In the Wave Load Condition combobox choose the wave load condition Condition1; see Figure 3.11. Click OK to define the analysis activity and the Activity Monitor appears; see Figure 3.12. Close the Activity Monitor dialog (it is reopened by Alt+D) and find Analysis1 in the Analysis > Activities folder. Click Analysis1 in the left pane of the browser (or double-click Analysis1 in the

right pane) to see the four activity steps: Meshing, Wave Load Analysis, Linear Structural Analysis and Load Results as shown in Figure 3.11.

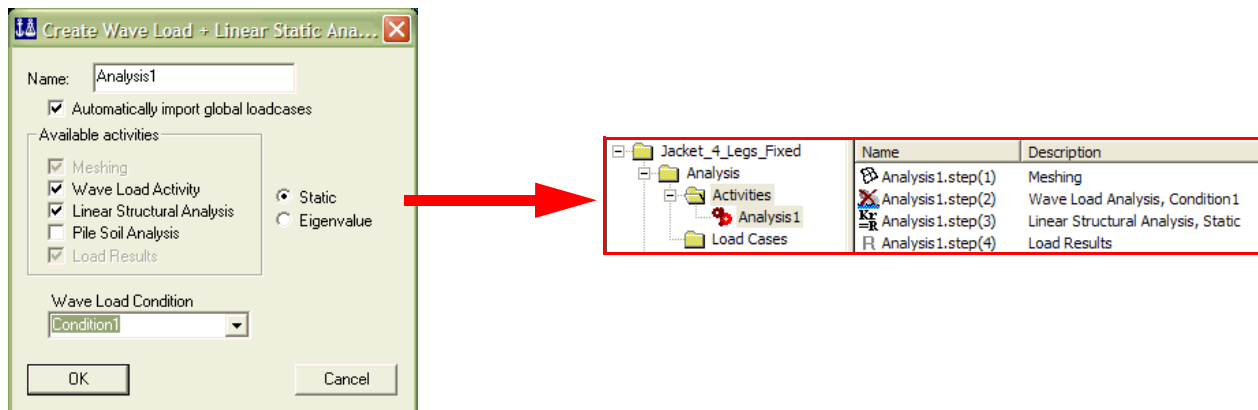


Figure 3.11 Analysis activities

**Note:** If you only want to run a wave load analysis (Wajac) then deselect Linear Structural Analysis in the dialog shown in Figure 3.11. Alternatively, you may deselect activities 1.3 and 1.4 in the Activity Monitor, see Figure 3.12.

## The Activity Monitor

The next step is to open the Activity Monitor dialog. Right-click the Analysis > Activities > Analysis1 object and select Activity Monitor. You can also do this by Tools > Analysis > Activity Monitor or simply by Alt+D. The Activity Monitor dialog, see Figure 3.12, appears with the same 4 steps as shown in Figure 3.11.

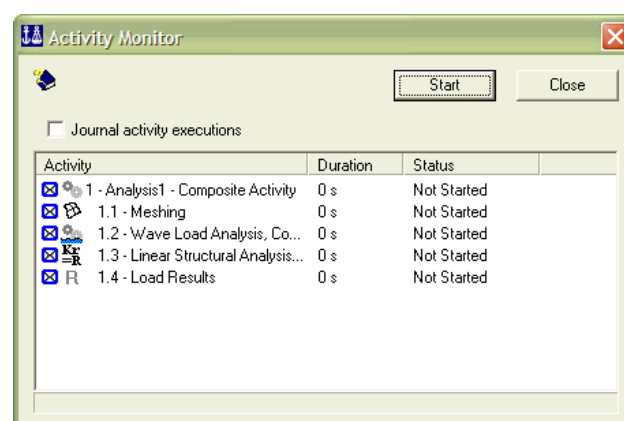


Figure 3.12 The Activity Monitor dialog before execution

Features of the Activity Monitor dialog are:

- Activities may be selected and deselected.



- Right-click an activity to edit the activity.
- Right-click an activity to access result files and other relevant actions.
- The Start button starts the execution.

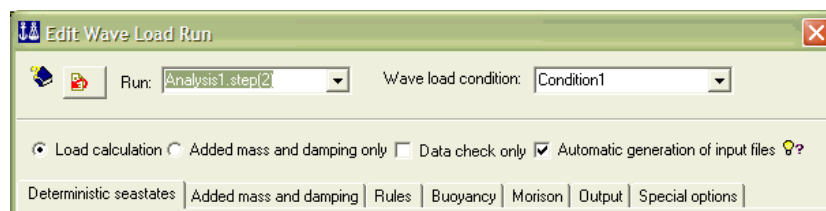
**Note:** Before starting the execution the definition of the Wave Load Analysis activity need to be completed.

### Edit Wave Load Run – complete the Wave Load Analysis activity

Open the Edit Wave Load Run dialog by right-clicking activity 1.2 - Wave Load Analysis, Condition1 and select Edit activity. The Edit Wave Load Run dialog appears; see Figure 3.13. The Run and Wave load condition combo boxes at top of the dialog are automatically filled in with appropriate data. We want to do a load calculation and not a data check only. Also, we want automatic generation of input files (for Wajac). Make sure these choices are made by the appropriate radio buttons and check boxes of the dialog.

All necessary run data is specified via the different tabs of the Edit Wave Load Run dialog. A more detailed description of the wave load analysis data is found in Section 5.4.

**Note:** The Automatic generation of input files check box, see Figure 3.13, is by default checked. This involves that the input to the wave load run, i.e. Wajac, is generated every time you run the analysis. Uncheck this box if you want to modify the input. Read more about this in Section 3.2.6.



**Figure 3.13** The Edit Wave Load Run dialog – upper part

### Deterministic seastates tab

The Phase step radio button is the default option and implies that phase stepping will be used for all waves; see Figure 3.14. The seastate table has five rows filled with values from the selected Wave Load Condition. The first four have wave and current specifications while the fifth is calm sea.

Now the empty cells in the table need to be filled in. For the first four rows, the waves, all parameters to enter will be the same. We want to store results for the load steps giving maximum base shear and maximum overturning moment; hence both Maximum base shear and Maximum overturning moment in the Design load calculation area are selected (see A in Figure 3.14). Other relevant data is specified in the fields of the Seastates table parameters area (B). Then the first four rows are selected by pressing and dragging the mouse over the rows (C). Conclude by clicking the Fill selected button (D). The empty cells of the first four rows are then filled. We want to compute wave loads without buoyancy, the Buoyancy column therefore need to be set to Off (E). The fifth row is for buoyancy calculation only, the Buoyancy column for this row therefore need to be set to On (F).

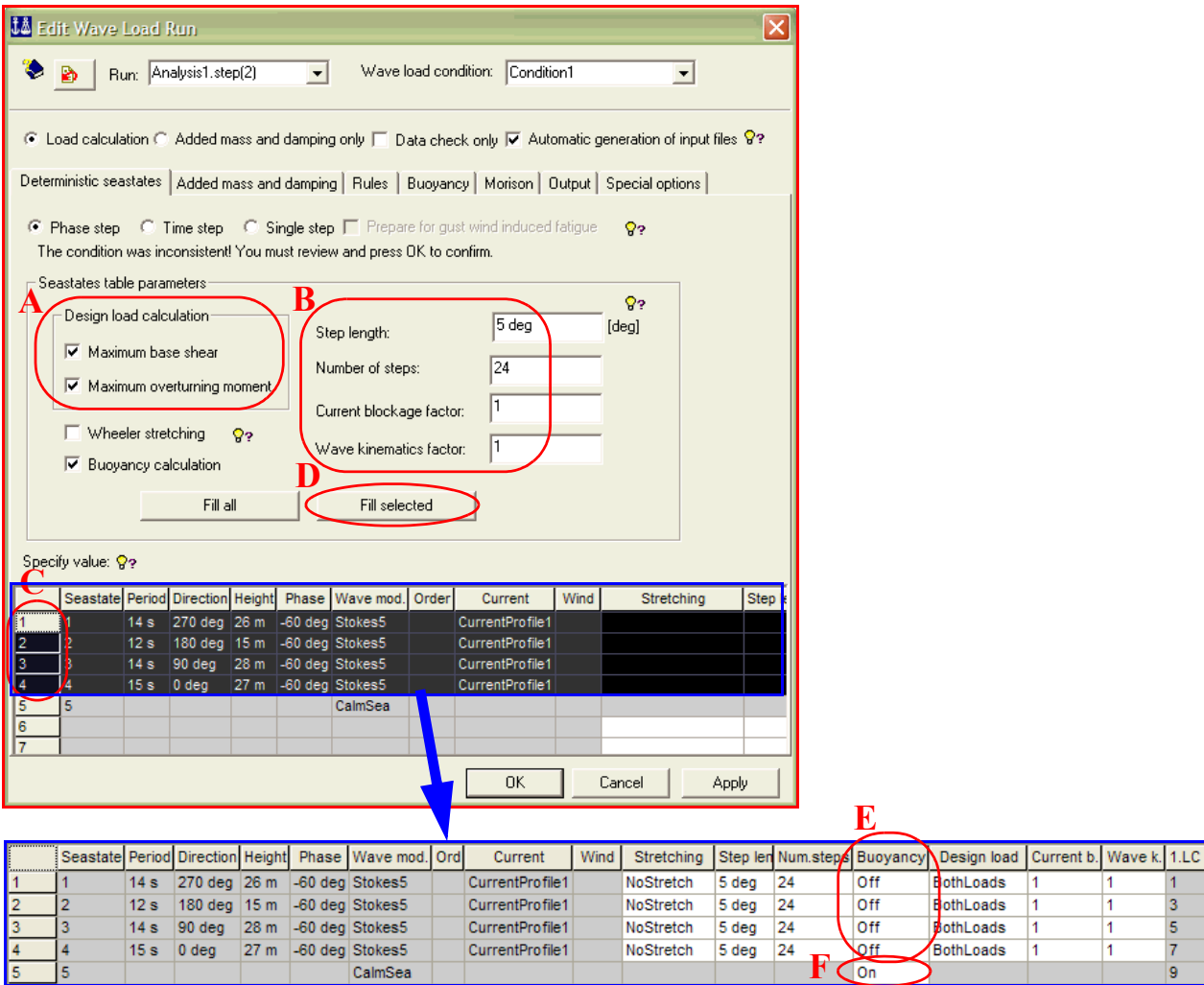


Figure 3.14 The Edit Wave Load Run > Deterministic seastates dialog

**Added mass and damping tab**

See Figure 5.44. This is irrelevant for a static analysis.

**Rules tab**

See Figure 5.45. Default values are used.

**Buoyancy tab**

See Figure 5.46. Default values are used.

**Morison tab**

See Figure 5.47. Default values are used.

**Output > Global results tab**

See Figure 5.48. Default values are used.

**Output > Print file tab**

See Figure 5.49. No extra print is specified.

**Output > Loads interface file tab**

See Figure 5.50. Default values are used.

**Special options tab**

See Figure 5.51. Default values are used.

**Complete Edit Wave Load Run dialog**

Having completed all tabs of the Edit Wave Load Run dialog click OK and the wave load cases appear under Analysis1 in the Analysis > Activities folder.

**3.2.4 Load Case names, Description, FEM Loadcase numbers and FEM LC Rule**

The hydrodynamic load case names are named <analysis activity name>.WLC(#) as shown in Figure 3.15.

**Note:** In Framework the hydrodynamic load case names appear as WLC(#).

Now create the eight load combinations of Table 3.1. Alternatively to Insert > Load Combination select the appropriate load cases under Analysis1 in the Analysis > Activities folder and right-click one of them to open the Insert Load Combination dialog. That way only the selected load cases appear in the dialog.

**Table 3.1 Load combinations**

<b>Load combination</b>	<b>Gravity FEM LC 1</b>	<b>Wind FEM LC 2-5</b>	<b>Wave FEM LC 6-13</b>	<b>Buoyancy FEM LC 14</b>
NorMaxSh	factor 1.2	WindN x 1.6	WLC(1) x 1.6	WLC(9) x 1
NorMaxMo	factor 1.2	WindN x 1.6	WLC(2) x 1.6	WLC(9) x 1
EasMaxSh	factor 1.2	WindE x 1.6	WLC(3) x 1.6	WLC(9) x 1
EasMaxMo	factor 1.2	WindE x 1.6	WLC(4) x 1.6	WLC(9) x 1
SouMaxSh	factor 1.2	WindS x 1.6	WLC(5) x 1.6	WLC(9) x 1

**Table 3.1 Load combinations**

<b>Load combination</b>	<b>Gravity FEM LC 1</b>	<b>Wind FEM LC 2-5</b>	<b>Wave FEM LC 6-13</b>	<b>Buoyancy FEM LC 14</b>
SouMaxMo	factor 1.2	WindS x 1.6	WLC(6) x 1.6	WLC(9) x 1
WesMaxSh	factor 1.2	WindW x 1.6	WLC(7) x 1.6	WLC(9) x 1
WesMaxMo	factor 1.2	WindW x 1.6	WLC(8) x 1.6	WLC(9) x 1

The combinations appear under Analysis1 in the Analysis > Activities folder.

To the right of the load case names and descriptions there are two columns: FEM Loadcase and FEM LC Rule. The former identifies the load case number that will appear in the Sestra listing (print file). The latter identifies the method or rule by which the load case number was assigned.

This works as follows:

- Basic load cases (created by Insert > Load Case) appear with FEM LC Rule set to Manual.
  - Basic load cases are by default numbered from 1 and up.
  - However, right-click > Properties on a basic load case may set its number to any free (high) number thereby leaving a gap in the numbering.
  - A new basic load case will be given the lowest free FEM Loadcase number. Unless there are gaps in the numbering this will be the highest basic load case number plus 1.
- Hydrodynamic load cases (wave loads) appear with FEM LC Rule set to Variable.
  - Hydrodynamic load cases are always numbered sequentially after the highest basic load case number. Creating a new highest basic load case number will therefore shift the hydrodynamic load case numbers correspondingly.
  - Hydrodynamic load cases are numbered in a sequence determined by the Edit Wave Load Run dialog.
  - Hydrodynamic load case numbers cannot be set by the user.
- Load combinations (created by Insert > Load Combination) appear by default with FEM LC Rule set to Automatic.
  - Load combinations are always numbered sequentially after the highest hydrodynamic load case number or, if there are no hydrodynamic load cases, after the highest basic load case number.
  - Load combinations are numbered according to an alphabetic sorting of their names. The sequence in which the load combinations were created is of no consequence.
  - Contrary to a hydrodynamic load case the number of a load combination may be set by the user. It can be set to any number higher than the highest hydrodynamic and basic load case number. Such setting changes its FEM LC Rule from Automatic to Manual. This in turn causes the remaining Automatic load combinations to be renumbered so as to follow sequentially this new highest numbered Manual load combination.

Figure 3.15 illustrates how an initial numbering of load cases is changed by (1) introducing a new load case (NewLoad) and (2) by setting the number of the first load combination to 21.

**Note:** The change in Figure 3.15 is for illustration purposes only and not a part of the tutorial.

Name	Description	FEM Loadcase	FEM LC Rule
R Analysis1.step(4)	Load Results		
Analysis1.step(2)	Wave Load Analysis, Condition1		
Analysis1.step(3)	Linear Structural Analysis, Static		
Analysis1.step(1)	Meshing		
Gravity	Reference to LoadCase	1	Manual
WindN	Reference to LoadCase	2	Manual
WindE	Reference to LoadCase	3	Manual
WindS	Reference to LoadCase	4	Manual
WindW	Reference to LoadCase	5	Manual
Analysis1.WLC(1)	Wave load condition	6	Variable
Analysis1.WLC(2)	Wave load condition	7	Variable
Analysis1.WLC(3)	Wave load condition	8	Variable
Analysis1.WLC(4)	Wave load condition	9	Variable
Analysis1.WLC(5)	Wave load condition	10	Variable
Analysis1.WLC(6)	Wave load condition	11	Variable
Analysis1.WLC(7)	Wave load condition	12	Variable
Analysis1.WLC(8)	Wave load condition	13	Variable
Analysis1.WLC(9)	Wave load condition	14	Variable
EasMaxMo	LoadCombination	15	Automatic
EasMaxSh	LoadCombination	16	Automatic
NorMaxMo	LoadCombination	17	Automatic
NorMaxSh	LoadCombination	18	Automatic
SouMaxMo	LoadCombination	19	Automatic
SouMaxSh	LoadCombination	20	Automatic
WesMaxMo	LoadCombination	21	Automatic
WesMaxSh	LoadCombination	22	Automatic

Name	Description	FEM Loadcase	FEM LC Rule
R Analysis1.step(4)	Load Results		
Analysis1.step(2)	Wave Load Analysis, Condition1		
Analysis1.step(3)	Linear Structural Analysis, Static		
Analysis1.step(1)	Meshing		
Gravity	Reference to LoadCase	1	Manual
WindN	Reference to LoadCase	2	Manual
WindE	Reference to LoadCase	3	Manual
WindS	Reference to LoadCase	4	Manual
WindW	Reference to LoadCase	5	Manual
NewLoad	LoadCase	6	Automatic
Analysis1.WLC(1)	Wave load condition	7	Variable
Analysis1.WLC(2)	Wave load condition	8	Variable
Analysis1.WLC(3)	Wave load condition	9	Variable
Analysis1.WLC(4)	Wave load condition	10	Variable
Analysis1.WLC(5)	Wave load condition	11	Variable
Analysis1.WLC(6)	Wave load condition	12	Variable
Analysis1.WLC(7)	Wave load condition	13	Variable
Analysis1.WLC(8)	Wave load condition	14	Variable
Analysis1.WLC(9)	Wave load condition	15	Variable
EasMaxMo	LoadCombination	21	Manual
EasMaxSh	LoadCombination	22	Automatic
NorMaxMo	LoadCombination	23	Automatic
NorMaxSh	LoadCombination	24	Automatic
SouMaxMo	LoadCombination	25	Automatic
SouMaxSh	LoadCombination	26	Automatic
WesMaxMo	LoadCombination	27	Automatic
WesMaxSh	LoadCombination	28	Automatic

Figure 3.15 FEM Loadcase numbers and FEM LC Rule when adding load and changing a number

3.2.5 Execution of the Design Wave Analysis

Now the execution of the design wave analysis may be started by clicking the Start button in the Activity Monitor. There is a Journal activity executions option that by default is not checked.

The progress of the execution along with time spent and the status at completion of each step is shown in the dialog. See Figure 3.16.

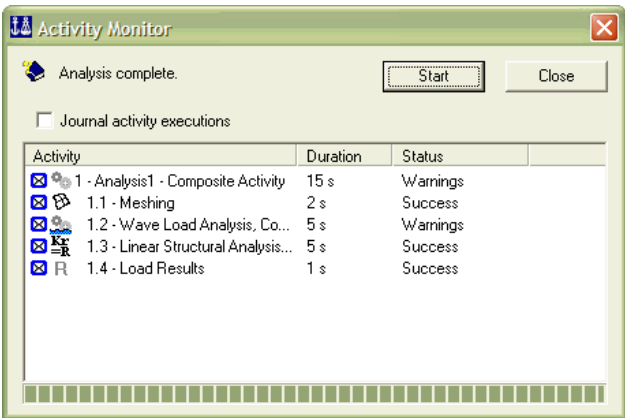
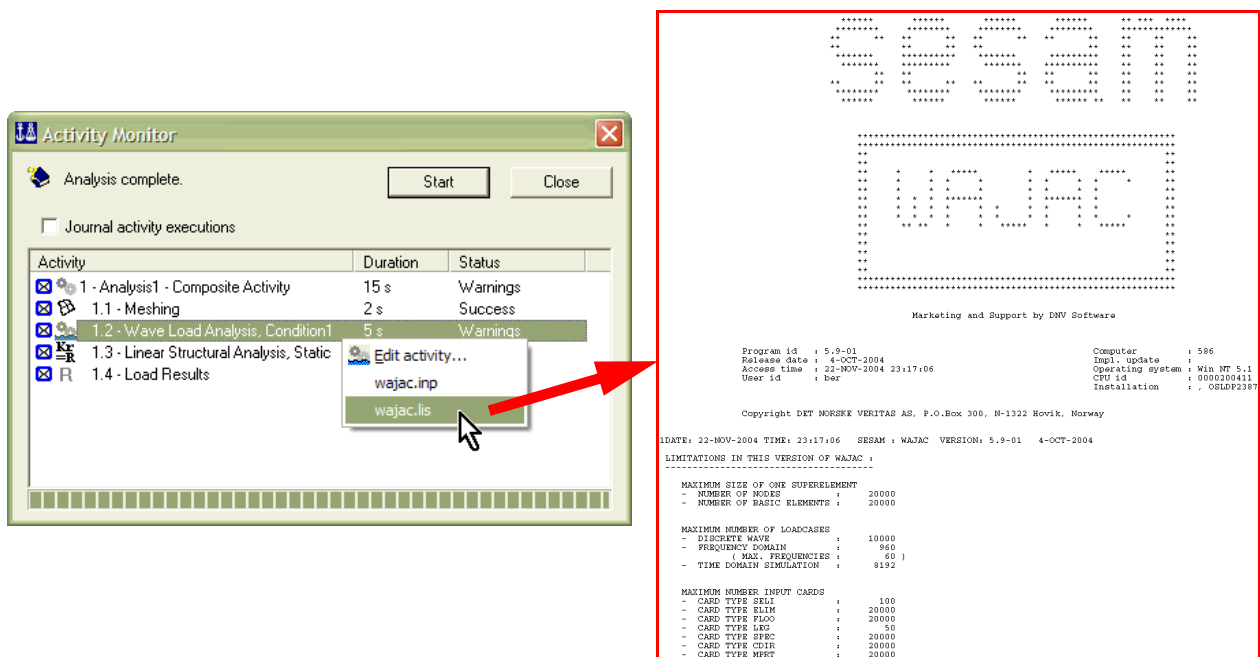


Figure 3.16 The Activity Monitor dialog after execution

After the execution inspect the hydrodynamic results found on the LIS-file (print file) of Wajac (see Section 4.2 about files used by GeniE) by right-clicking the Wave Load Analysis activity and selecting `wajac.lis`; see Figure 3.17. Also the Wajac input file may be inspected in the same way. The files appear in the text editor associated with the file type. Read more about this in Section 5.5.



**Figure 3.17** Inspect the Wajac LIS-file via right-click in the Activity Monitor dialog

When the analysis has successfully been completed the results may be graphically inspected.

First switch the display to Results - All and then use Tools > Analysis > Presentation (or Alt+D) to present for example Beam Forces - NXX for load combination SouMaxMo as shown in Figure 3.18.

26 Sep 2005 11:27  
Jacket\_4\_Legs\_Fixed  
Analysis1  
SouMaxMo  
FEM Load Case = 19  
Beam Forces - Nxx  
Min: -120227  
Max: 45786.7

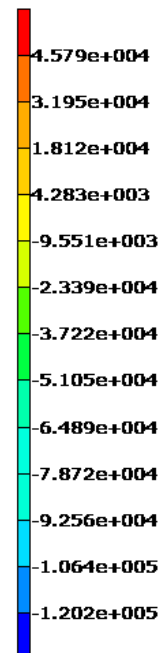
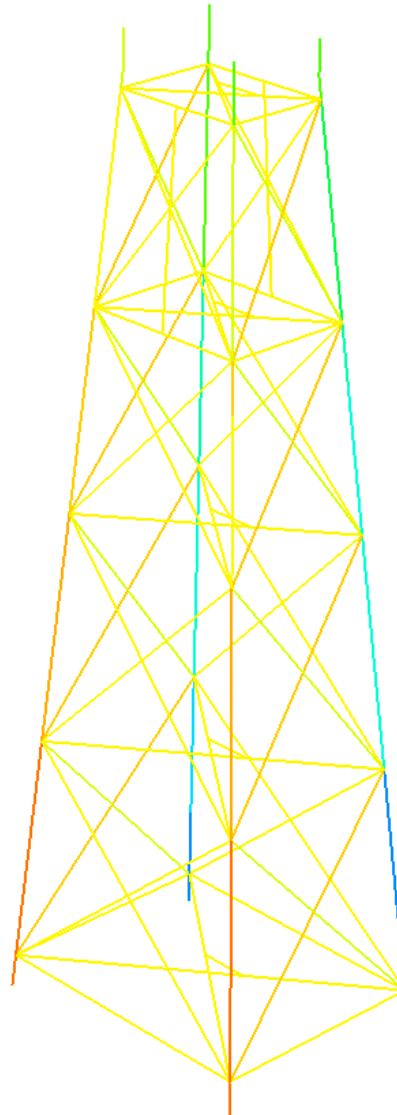
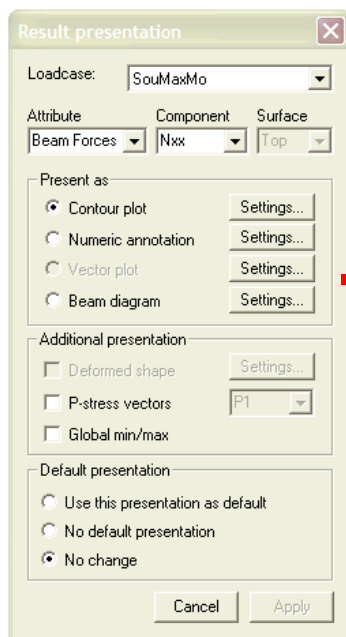


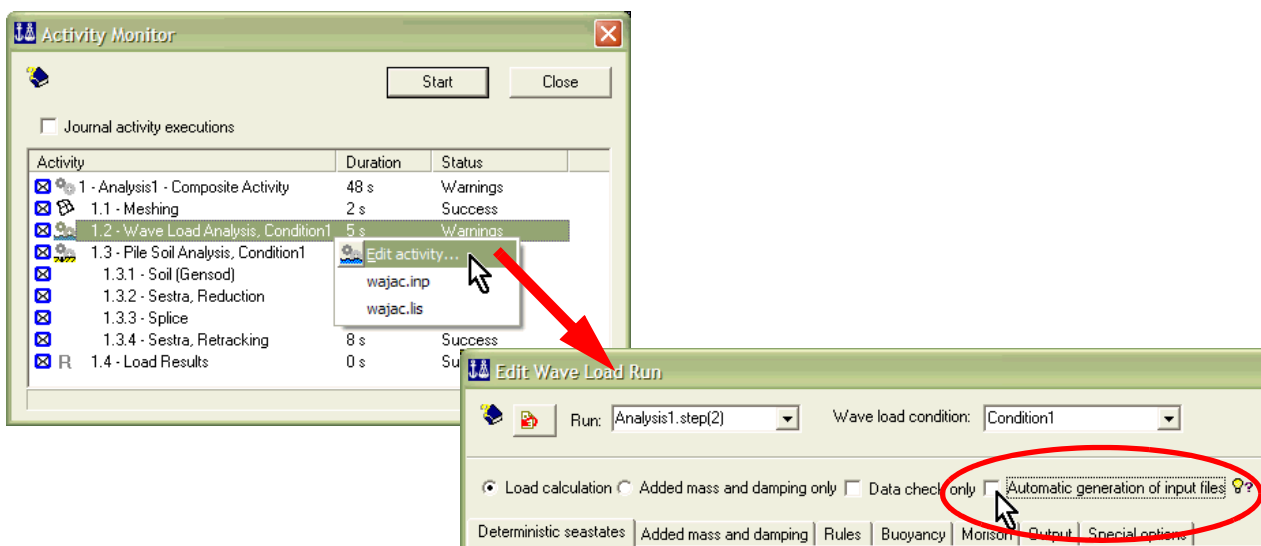
Figure 3.18 Axial beam force results from a static analysis

### 3.2.6 Execution of the Design Wave Analysis with Modified Input

You may want to modify the wave analysis (Wajac) input. This may be required in order to give input outside the capabilities of GeniE.

Do so by opening the Activity Monitor (Alt+D), right-clicking activity '1.2 - Wave Load Analysis, Condition1', selecting Edit activity, unchecking Automatic generation of input files (see Figure 3.19) and closing the dialog by OK. Thereafter, right-click the activity again and select wajac.inp – the input file to Wajac. The file appears in the text editor associated with the file type. Edit the input as required and save the new version of the file using the same file name.

Click the Start button in the Activity monitor to start the revised execution.



**Figure 3.19 Uncheck Automatic generation of input files in order to manually edit input**

**Note:** As long as Automatic generation of input files is unchecked changes to hydro property data and other data controlled by GeniE will not take effect in an analysis. When such changes are made in GeniE remember to check Automatic generation of input files. In turn, you may then need to uncheck it again and redo the manual changes to the Wajac input file.



### 3.3 Getting Started doing Deterministic Fatigue Wave Analysis

This is a description of how a deterministic fatigue wave analysis is performed. Input of environment, hydro properties and wave load run data is described. Much of the input is similar to the input in Section 3.2 Getting Started doing Design Wave Analysis. Focus is therefore put on what is different from the design wave analysis.

If you want to perform the operations described below then go to Section A 2 for the input of the structural data. (The input may be copied from an electronic version of the manual.)

Normally, a deterministic wave load analysis consists of a number of wave components. In this simplified example, however, there are only a few.

Four wave headings will be analysed. For each wave heading there are four waves with different pairs of wave periods and wave heights. The set of periods and heights are the same for all wave directions. We then have a total of  $4 \times 4 = 16$  number of seastates. Linear waves (Airy) will be used and we have no current and no wind.

#### 3.3.1 Environment Data

The following environment data are defined: a location (similar to the design wave analysis in Section 3.2), a regular wave set and a wave load condition.

##### Define the Location

See Define the Location in Section 3.2.1.

##### Define the Waves

A Regular Wave Set constituted by wave components is defined. Each component consists of period, height, phase angle and direction data. The phase angles will serve as initial phase angles in the seastates defined in the Edit Wave Load Run dialog (Figure 5.41).

Right-click the Environment > Water folder and select New Regular Wave Set (Figure 5.15). Select the Wave Set Type Period radio button. Values for periods and heights are entered directly into the table.

**Note:** Values may also be copied from a spreadsheet, e.g. Excel.

To fill the rest of the table the Fill tools area is used. The initial phase angle is 0 for all wave components so the Phase radio button is selected and the value 0.0 is entered into the Value edit box in the Single value area; see Figure 3.20. The Fill all button is clicked.

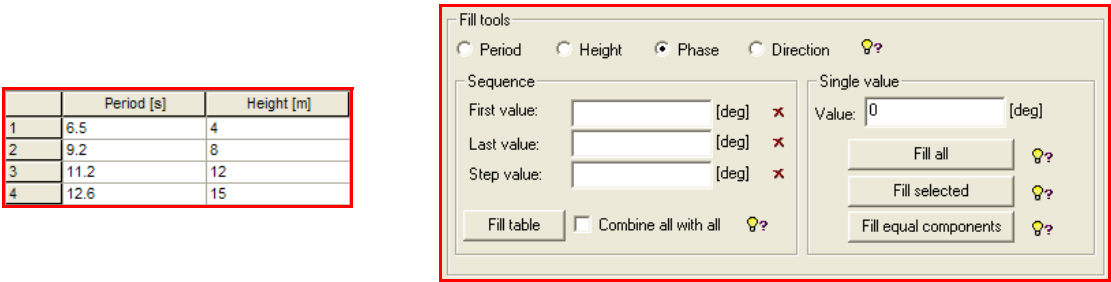
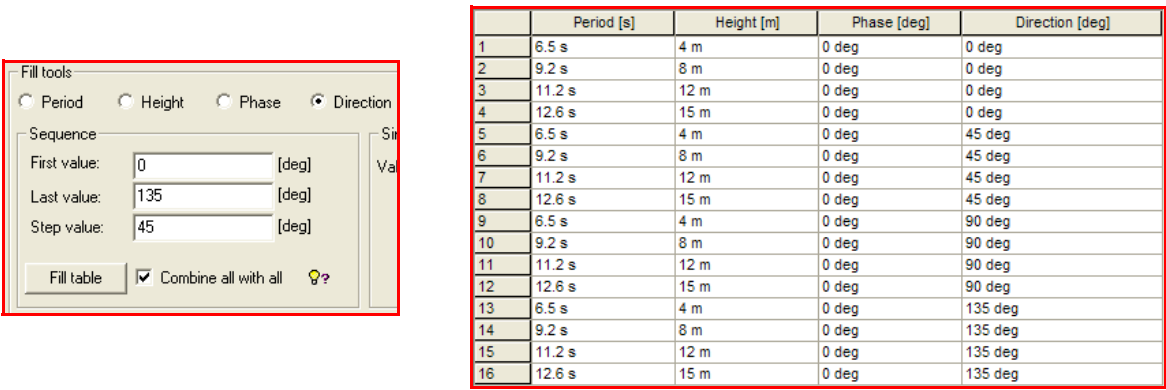


Figure 3.20 Define wave set – wave periods, heights and phases

To complete the wave components table the directions must be filled in. The directions are 0, 45, 90 and 135. Select the Direction radio button and in the Sequence area enter the first, last and step values. Tick the Combine all with all check box. Click the Fill table button and a total of 16 wave components appear in the wave components table as shown in Figure 3.21. Click OK to accept the data and close the dialog.



	Period [s]	Height [m]	Phase [deg]	Direction [deg]
1	6.5 s	4 m	0 deg	0 deg
2	9.2 s	8 m	0 deg	0 deg
3	11.2 s	12 m	0 deg	0 deg
4	12.6 s	15 m	0 deg	0 deg
5	6.5 s	4 m	0 deg	45 deg
6	9.2 s	8 m	0 deg	45 deg
7	11.2 s	12 m	0 deg	45 deg
8	12.6 s	15 m	0 deg	45 deg
9	6.5 s	4 m	0 deg	90 deg
10	9.2 s	8 m	0 deg	90 deg
11	11.2 s	12 m	0 deg	90 deg
12	12.6 s	15 m	0 deg	90 deg
13	6.5 s	4 m	0 deg	135 deg
14	9.2 s	8 m	0 deg	135 deg
15	11.2 s	12 m	0 deg	135 deg
16	12.6 s	15 m	0 deg	135 deg

Figure 3.21 Define wave set – wave directions and resulting components

Define the Wave Load Condition

The Wave Load Condition defines how the basic environmental data will be used in the analysis. A Condition of type Deterministic Time is created. Right-click the Environment > Location1 icon and select New Wave Load Condition. The Deterministic Time radio button is currently the only option.

Tick the Regular wave set check box and select the previously defined WaveSet1 from the combobox in the Wave components area. The wave components appear in the table as shown in Figure 3.22.

Use the Assign wave component properties area to complete the table. Select Airy as Wave model and click the Fill All button. The table is now complete. Click OK.

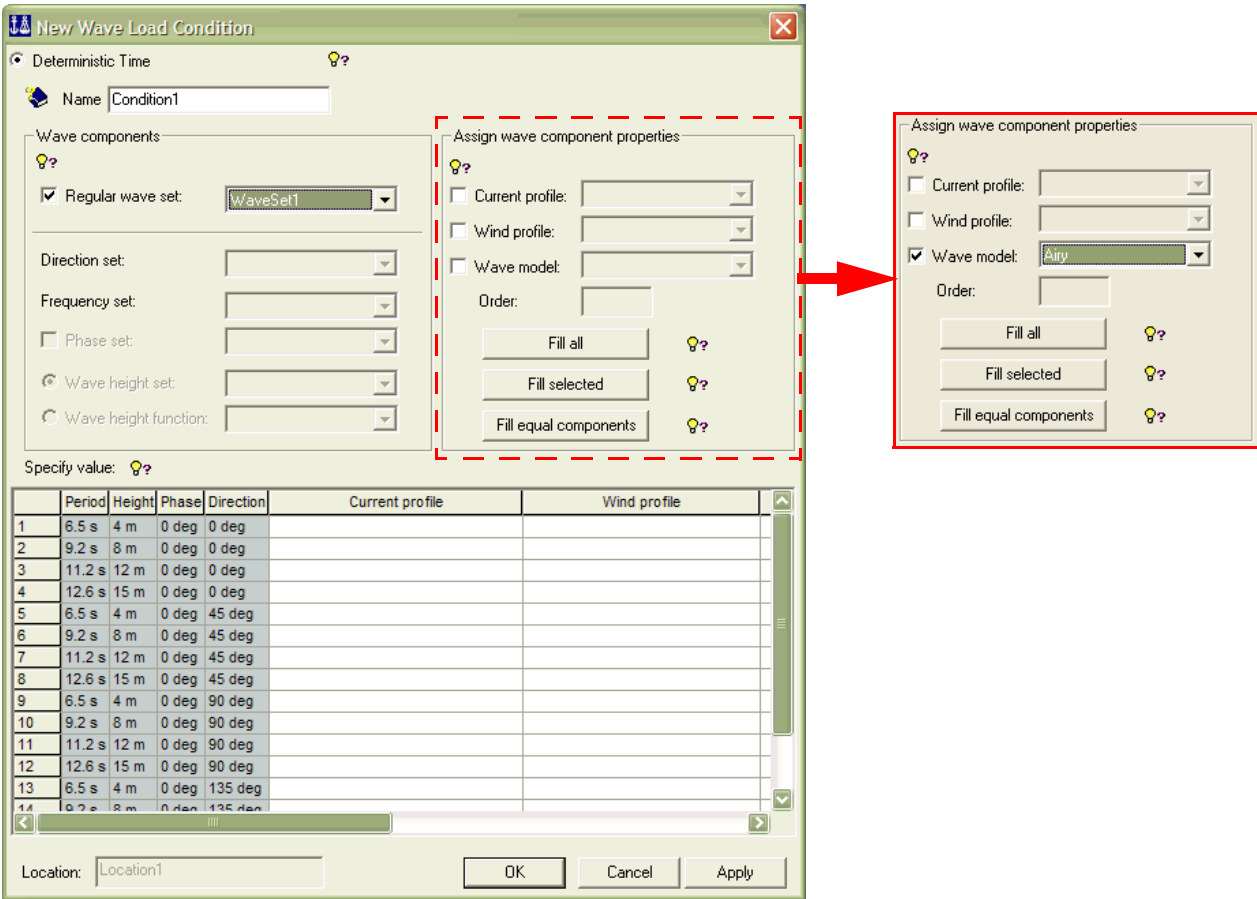


Figure 3.22 The New Wave Load Condition dialog

### 3.3.2 Hydro Properties

The hydro properties to be specified are Morison coefficients, Marine Growth, Flooding and Element refinement.

#### **Define and assign Morison Coefficients**

See Define and assign Morison Coefficients in Section 3.2.2.

#### **Define and assign Marine Growth**

See Define and assign Marine Growth in Section 3.2.2.

#### **Define and assign Flooding**

See Define and assign Flooding in Section 3.2.2.

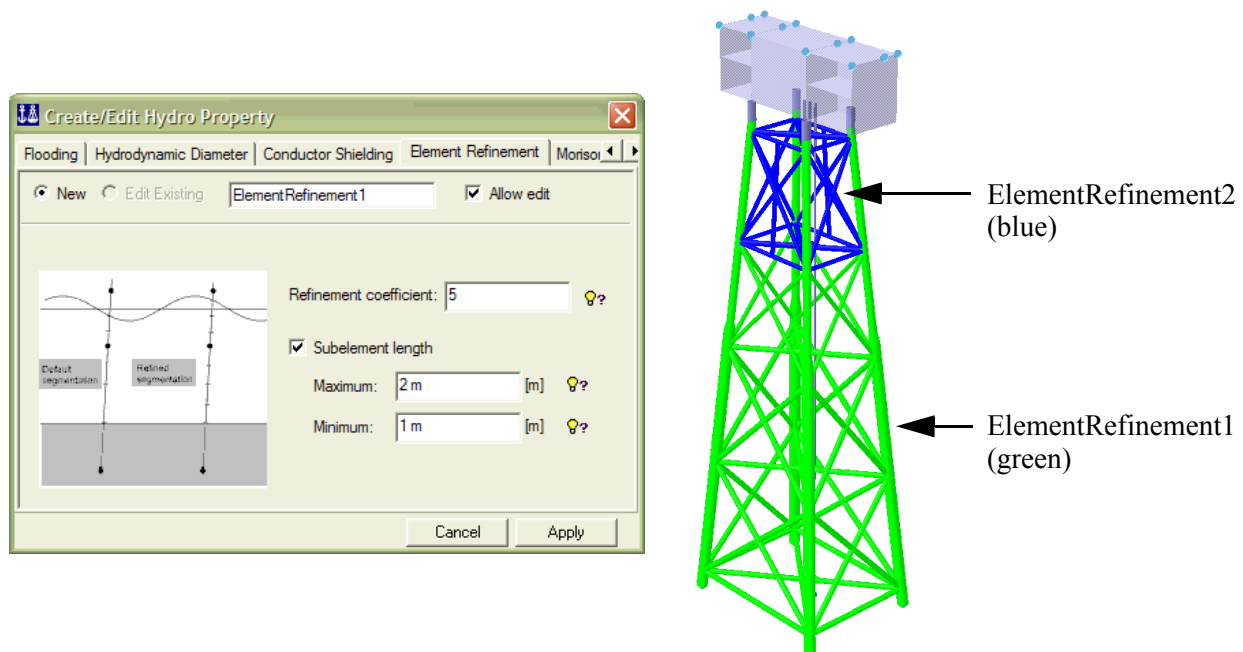
#### **Define and assign Element Refinement**

An Element Refinement property is defined and assigned to the structure. In this example two properties are defined. ElementRefinement1 is assigned to the bracing in the interval from the seabed and up to Z=105 as well as to the whole length of the legs. ElementRefinement2 is assigned to the bracing in the interval from (and including) Z=105 to the water surface. For a detailed description of this property see text related to Figure 5.36.

Right-click the Properties > Hydro folder and select New Element Refinement. The Element Refinement tab appears; see Figure 3.23. (You may also do this by Edit > Properties > Hydro Property tab > Create/Edit Hydro Property > Element Refinement.) Enter Refinement coefficient = 5, Maximum Subelement length = 2 and Minimum Subelement length = 1. Connect the ElementRefinement1 property to the relevant parts of the structure.

Repeat the process by defining ElementRefinement2 with Refinement coefficient = 15, Maximum Subelement length = 2 and Minimum Subelement length = 1. Connect the ElementRefinement2 property to the relevant parts of the structure.

Normally, high refinement is preferred close to the water surface where the variation in wave loads is high and low refinement is preferred close to seabed. Ensure that you have previously defined named sets of relevant parts of the structure. The refinement property may optionally be skipped since Wajac handles the refinement by default.



**Figure 3.23 The Create/Edit Hydro Property > Element Refinement dialog**

### 3.3.3 Define the Deterministic Fatigue Wave Analysis

The comments in Section 3.2.3 for how to create a Wave Load Analysis are also relevant here.

#### The Activity Monitor


See Section 3.2.3 about the Activity Monitor. Before starting the execution complete the definition of the Wave Load Analysis activity. Open the Edit Wave Load Run dialog by right-clicking activity 1.2 and selecting Edit activity.

#### Edit Wave Load Run – complete the Wave Load Analysis activity

All necessary run data are specified through the choices at top and the various tabs of the Edit Wave Load Run dialog. A more detailed description of the wave load analysis data is found in Figure 5.4.

#### Deterministic seastates tab (Figure 5.43)

As in the design wave analysis in Section 3.2 the Phase step option is used. The seastates table has 16 rows filled with values from the selected Wave load condition Condition1. To fill the empty cells of the table the Seastates table parameters area is used. In this example the step length is 30 degrees and number of steps is 12. The factors for current blockage and wave kinematics are both 1.0. The wave theory used is linear (Airy) with Wheeler stretching of the fluid kinematics. No Buoyancy calculation is wanted. Since all states shall have the same values the Fill all button is clicked. The white cells are then filled with values. The last column shows the loadcase number for the first loadcase for each wave component.

☒ Phase step
 ☐ Time step
 ☐ Single step
 ☐ Prepare for gust wind induced fatigue
 


The condition was inconsistent! You must review and press OK to confirm.


Seastates table parameters

Design load calculation

☐ Maximum base shear

☐ Maximum overturning moment

☒ Wheeler stretching 

☐ Buoyancy calculation 

Step length:  [deg]

Number of steps:

Current blockage factor:

Wave kinematics factor:

	Seastate	Period	Direction	Height	Phase	Wave	Stretching	Step length [deg]	Num.steps	Buoyancy	Design load	Current b	Wave k. fac.	1.LC num.
1	1	6.5 s	0 deg	4 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	1
2	2	9.2 s	0 deg	8 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	13
3	3	11.2 s	0 deg	12 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	25
4	4	12.6 s	0 deg	15 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	37
5	5	6.5 s	45 deg	4 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	49
6	6	9.2 s	45 deg	8 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	61
7	7	11.2 s	45 deg	12 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	73
8	8	12.6 s	45 deg	15 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	85
9	9	6.5 s	90 deg	4 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	97
10	10	9.2 s	90 deg	8 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	109
11	11	11.2 s	90 deg	12 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	121
12	12	12.6 s	90 deg	15 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	133
13	13	6.5 s	135 deg	4 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	145
14	14	9.2 s	135 deg	8 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	157
15	15	11.2 s	135 deg	12 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	169
16	16	12.6 s	135 deg	15 m	0 deg	Airy	WheelerStretc	30 deg	12	Off	NoDesignLoads		1	181

Figure 3.24 Edit wave load activity – Deterministic seastates tab

**Added mass and damping tab**

See Figure 5.44. This is irrelevant for a static analysis

**Rules tab**

See Figure 5.45. Default values are used.

**Buoyancy tab**

See Figure 5.46. Default values are used.

**Morison tab**

See Figure 5.47. Default values are used.

**Output > Global results tab**

See Figure 5.48. Default values are used.

**Output > Print file tab**

See Figure 5.49. No extra print is specified.

**Output > Loads interface file tab**

See Figure 5.50. Default values are used.

**Special options tab**

See Figure 5.51. Default values are used.

When the Wave Load Analysis definition is complete the wave loads appear under Analysis1 in the Analysis > Activities folder.

**Note:** The waves may be visualised in the graphical view when a wave load case is selected by the set-Current command. Currently only Airy type waves have this feature. An example is shown in Figure 3.25.

23 Nov 2004 01:28  
Jacket\_4\_Legs\_DetFat  
Analysis1.WLC(9)

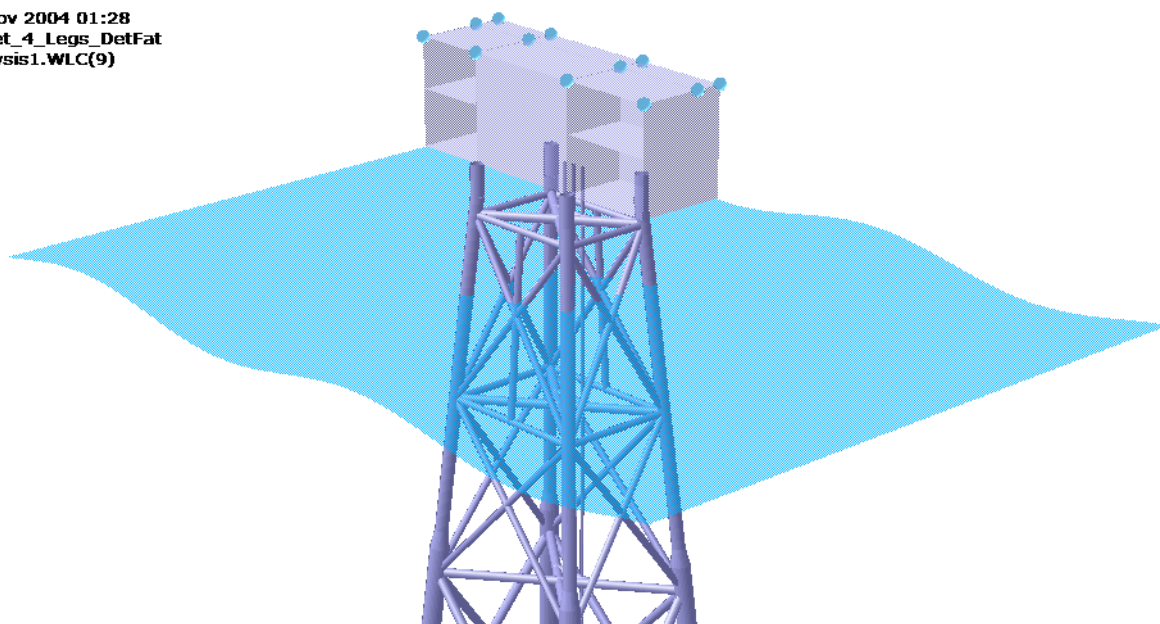


Figure 3.25 Airy wave displayed with model

### 3.3.4 Execution of Deterministic Fatigue Wave Analysis

The comments in Section 3.2.5 for how to execute a wave load analysis are also relevant here.

## **3.4 Getting Started doing Stochastic (Spectral) Fatigue Wave Analysis**

Stochastic (spectral) fatigue wave analysis is not offered by the 3.2 version of GeniE. It will, however, be offered in a coming version of GeniE.

Currently, such analysis is performed by exporting the FEM-file from GeniE by Tools > Analysis > Export FEM File (see Section 4.2 about files used by GeniE) and controlling the analysis from SESAM Manager instead. See Section 3.7 about how to do analyses outside GeniE.

## **3.5 Introduction to Structure-Pile-Soil Analysis**

### **3.5.1 Concepts in a Structure-Pile-Soil Analysis**

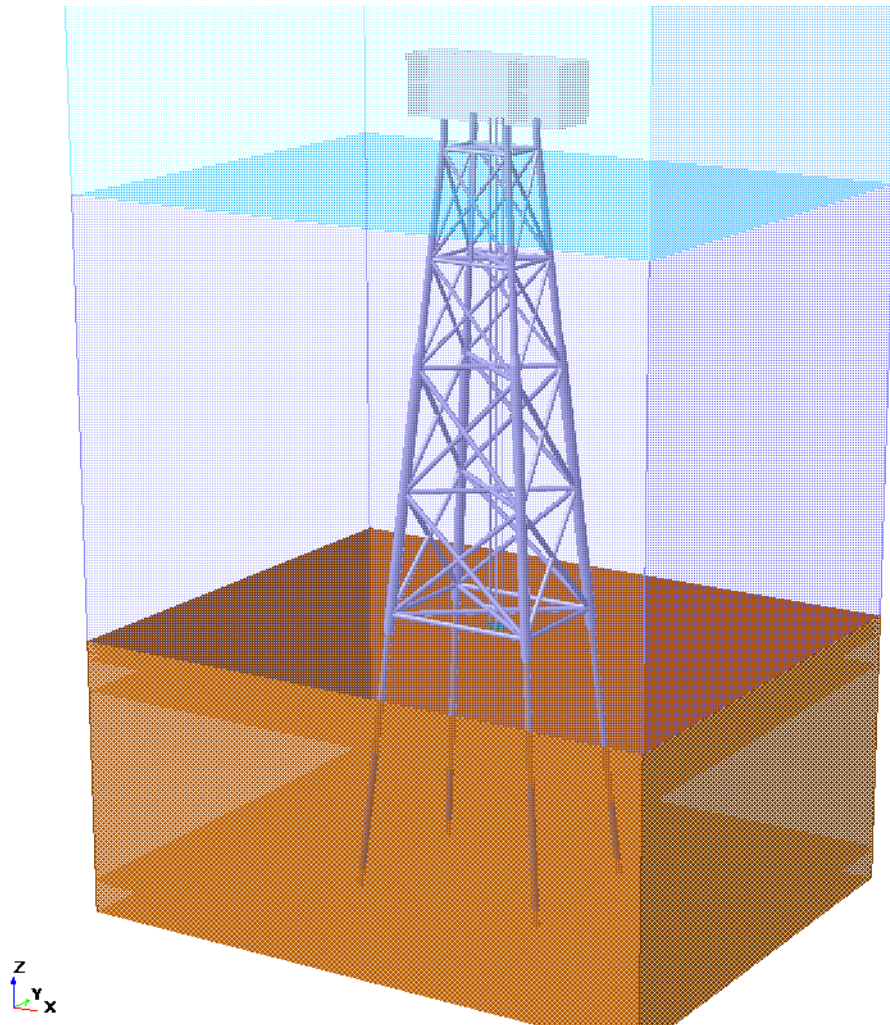
The main concepts in a structure-pile-soil model are:

- The environment model including soil layers and their properties in addition to wind and waves
- The structure model, e.g. a jacket
- The piles with associated properties

#### **Environment Model**

The environment model is referred to as a location. A location describes the relevant information about a site. It typically contains information about soil, water, waves and air; see Figure 3.26. In the following emphasis is put on describing the soil layers. Such an analysis will in most cases also involve a wave load analysis but this is covered by Section 3.1 through Section 3.3.





**Figure 3.26 Location in air, water and soil**

### Structure Model

The structure model can be any type of model but usually it is a jacket model consisting of beams with pipe cross sections, optionally including a topside structure with stiffened plates.

### Piles

In a structure-pile-soil analysis the structure is supported by piles that interact non-linearly with the soil. GeniE provides specific pile concepts for this purpose. A pile is similar to a beam but it has special additional properties such as pile tip information. A pile also relates directly to soil layers and sublayers in such a way that the division of the pile into nodes and elements is a direct function of the soil layer description.

### Structure-Pile-Soil Analysis

A complete structure-pile-soil analysis in GeniE consists of several steps:

- Meshing of structure and piles
- Execution of wave load analysis, creating the wave loads
- Combination of static loads and wave loads into load combinations representing complete loading conditions
- Linear reduction of structure stiffness model
- Non-linear solution of pile-soil interaction
- Retracking of forces and moments in structure and pile model

This process is automated as a Pile-Soil activity. An activity involves use of several externally controlled applications including Wajac, Gensod and Splice. The activity is retained in the GeniE database and can be executed many times. It is possible to edit and modify input to the externally controlled applications via the activity.

### 3.5.2 Modelling

In a structure-pile-soil analysis the task of modelling consists of two main parts: modelling the environment and modelling the piles and structure.

Since the piles relate to and depend on the soil layers it is natural to start modelling the environment and the soil layers. The soil layers (and sublayers) are directly used in the pile meshing to control the position of pile nodes.

#### Soil Modelling

Start by creating a new location object containing soil, water and air zones. Essential parts of the soil data include the seabed Z-coordinate (mudline) and the bottom soil layer Z-coordinate.

For the seabed, one may e.g. enter data about general and local scour; see Figure 3.27. For the soil layers, one may choose between clay and sand followed by entering soil data and soil curve properties; see Figure 3.28. These data control the generation of P-Y, T-Z and Q-Z curves.

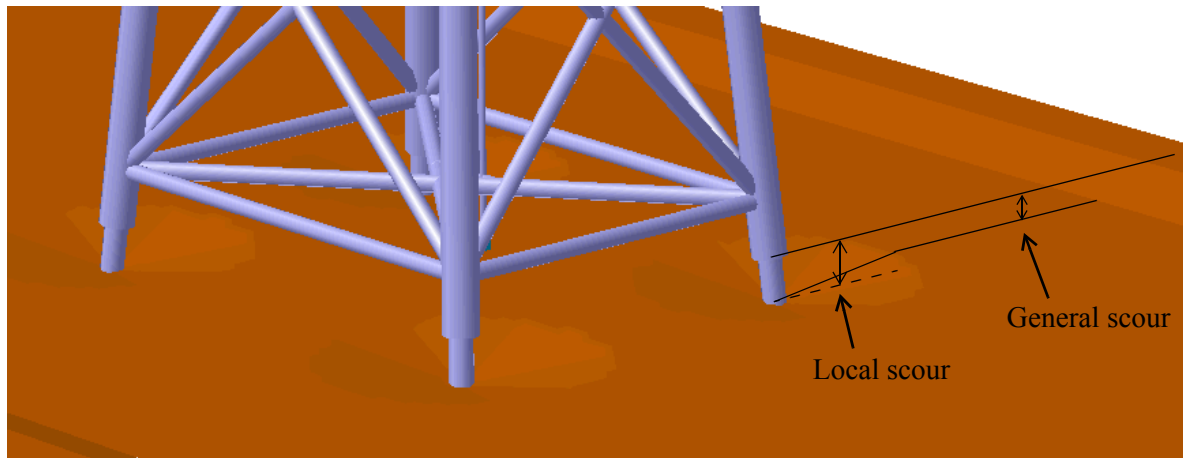


Figure 3.27 General and local scour

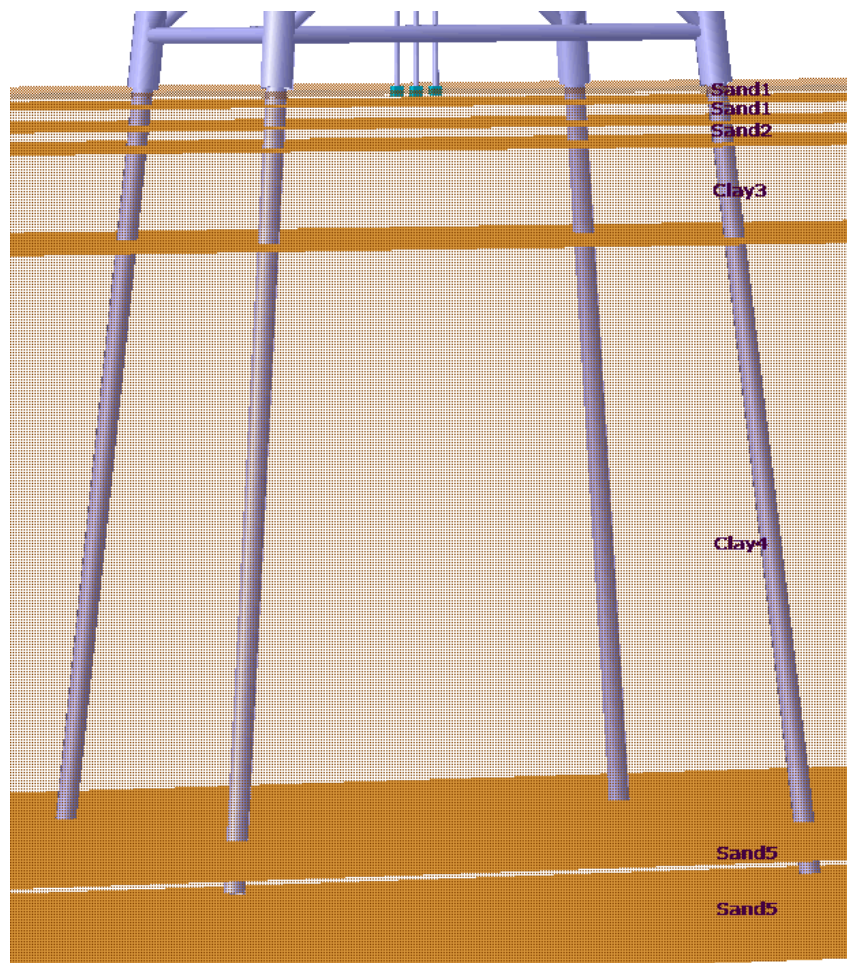


Figure 3.28 Soil layers

It is also possible to specify sublayers within each soil layer. The purpose of the sublayers is to control the discretisation of the piles, i.e. the number of nodes and elements along the piles when meshed. Typically, one may want to use more sublayers near the seabed.

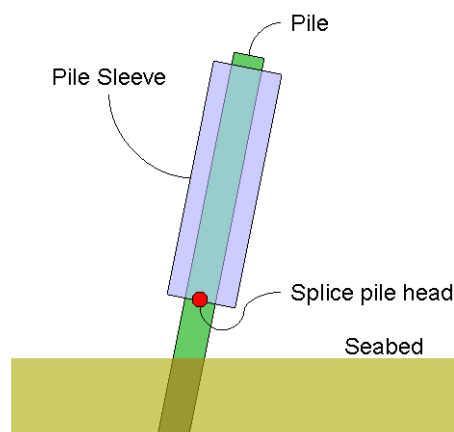
### Pile Modelling

After modelling the location it is natural to model the structure and piles (it is of course possible to model the structure first).

The bottom part of the structure, not including the piles, is typically at or near the seabed Z-coordinate.

Below the seabed piles should be modelled using the special pile concept provided. Only piles modelled this way will be subject to non-linear pile-soil analysis in Gensod and Splice.

It is possible for piles to overlap beams in the jacket structure in order to represent pile-in-sleeve and pile-in-leg designs. The pile sleeve or leg is modelled as an ordinary beam with pipe cross section overlapping the pile. In Splice only the lower non-overlapping part of the pile is considered. The 'Splice pile head' is located at the bottom of the pile sleeve or at the top of the pile if there is no pile sleeve; see Figure 3.29.



**Figure 3.29 Pile in pile sleeve**

A pile is easily modelled by the following three step procedure (the letters refer to Figure 3.40):

- a Select the pile top (click a snap point in the structure, e.g. the lower end of a leg or pile sleeve)
- b Specify the Z-level of the pile tip in the Snap plane dialog opening up (do not click anywhere in the model)
- c Align the pile by clicking an alignment beam in the structure, e.g. a leg or pile sleeve

### Analysis Parameters

Once the modelling is complete a structure-pile-soil analysis Activity can be created in which control information for the analysis can be accessed and modified.

### **3.5.3 Analysis**

The structure-pile-soil analysis activity controls everything from creating a finite element model to the equation solving and results retrieval.

#### **Representation**

The structure, modelled by beams, is meshed as two node beam elements in the finite element model. The piles are also meshed as two node beams thereby enabling presentation of bending moment diagrams for the piles after the analysis.

The structure and piles belong to the same 1st level superelement. All nodes along the piles are automatically defined as supernodes. There is also a supernode-coupling between the pile heads and the bottom of the structure (bottom of leg or pile sleeve). GeniE automatically creates a 2nd level superelement and ‘lifts’ (‘combines’ in Presel terminology) all loadcases and load combinations to the 2nd level superelement. When the analysis has been completed the results are presented for the 1st level superelement, i.e. the structure model. This way, a seamless structure-pile-soil analysis is enabled without the user being concerned with superelement modelling, assembling and analysis. A superelement analysis is performed in the background hidden for the user.

#### **Execution**

The structure-pile-soil analysis is completely integrated. There is no need for leaving GeniE.

### **3.5.4 Results**

For Pile-Soil analyses the following results may be presented in GeniE:

- The deformed shape of structure and piles
- Bending moment, shear force and axial force diagrams
- More specialised results are available in the Gensod and Splice listing files that are easily accessed from the GeniE Activity Monitor.

### 3.6 Getting Started doing Structure-Pile-Soil Analysis with Design Wave Loads

A four legged jacket with a pile in the extension of each leg is used as an example. The soil is divided into 7 layers with different properties. The wave loads consist of 4 design waves with different directions, periods and wave heights.

If you want to perform the operations described below then go to Section A 3 for the input of the structural data. (The input may be copied from an electronic version of the manual.)

#### 3.6.1 Environment

Start by creating a new location. Right-click the Environment folder and select New Location. (You may also do this by Insert > Environment > Location.) The Create/Edit Location dialog appears with default values. Set Waterline Z found in the Water tab to 124 m and Seabed Z found in the Soil tab to 0 m; see Figure 3.30. Click OK in the dialog.

**Note:** These values are given in the model (global) coordinate system.

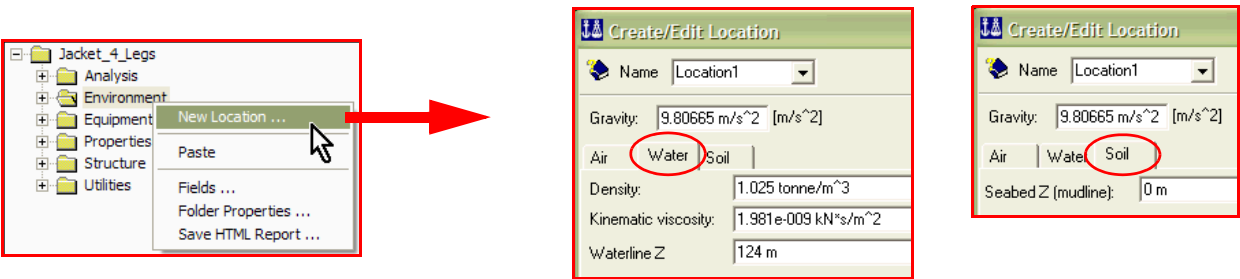


Figure 3.30 Create a new location and edit it

When the location is created a Location icon appears in the Environment folder (see Figure 3.31) and the water and soil layers are displayed. The visibility and layout is modifiable using the Location selection button in the toolbar.

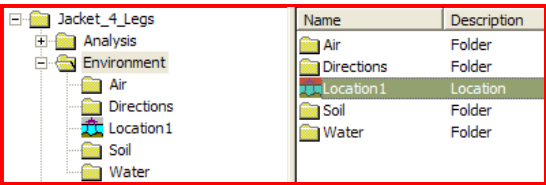


Figure 3.31 Location in Environment folder

### 3.6.2 Scour

First define the scour by right-clicking the Environment > Soil folder and selecting New Scour. Fill in data for general and local scour as shown in Figure 3.32. Click Apply (and Cancel).

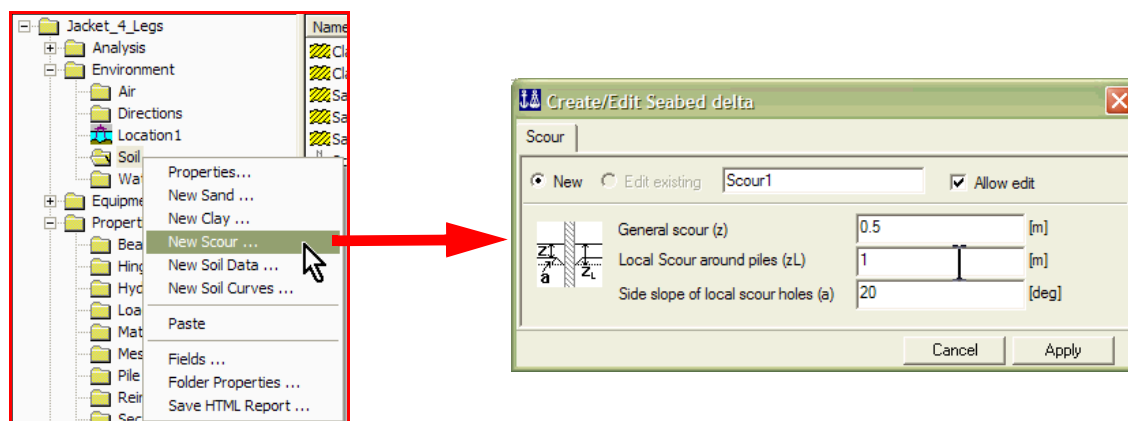


Figure 3.32 The Create/Edit Seabed delta dialog defines scour

### 3.6.3 Soil Properties

In order to create the soil there are four types of property that need to be defined:

- Soil Type is a property with soil type and principal characteristics.
- Soil Curves is a property that controls the generation of P-Y, T-Z and Q-Z curves.
- Soil Data is a property with additional soil characteristics.
- Soil layers and their properties.

Reopen the Create/Edit Location dialog for editing by right-clicking Location1 in the browser and go to the Soil tab. In the Seabed delta field specify the seabed delta type Scour1; see Figure 3.33.

Define new soil types via the option <Create new Soil> as shown in Figure 3.33 or by right-clicking the Environment > Soil folder as shown for scour in Figure 3.32. Figure 3.34 shows the dialogs for creating soil of types Sand and Clay. Figure 3.35 provides all necessary soil data. First create soil types Soil1 through Soil5 with data provided by the two tables in the upper left of Figure 3.35. Thereafter, fill in the table in the Create/Edit Location dialog with soil layer Z bottom and soil type.

Use a similar approach for soil curves and soil data. A single soil curve is defined and used for all layers; see the dialog to the left in Figure 3.36. The soil data to enter in the dialog shown to the right in Figure 3.36 is provided by the table at the bottom of Figure 3.35. Having created the soil curve and soil data fill in the Create/Edit Location dialog as shown in Figure 3.36 and click OK to accept the data.

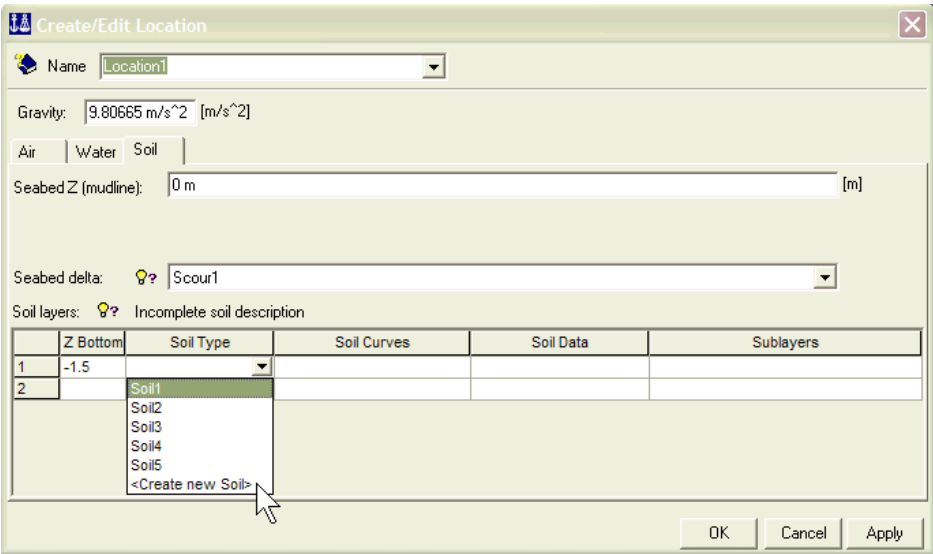


Figure 3.33 The Create/Edit Location dialog

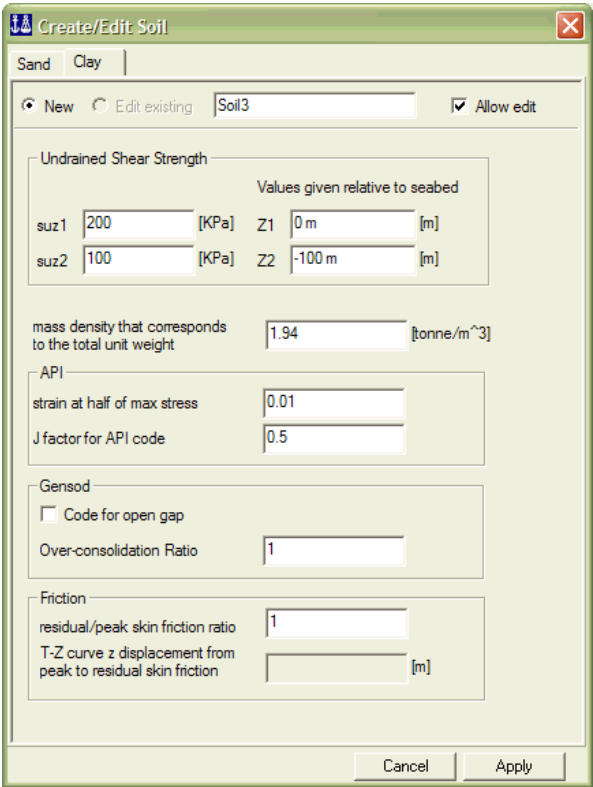
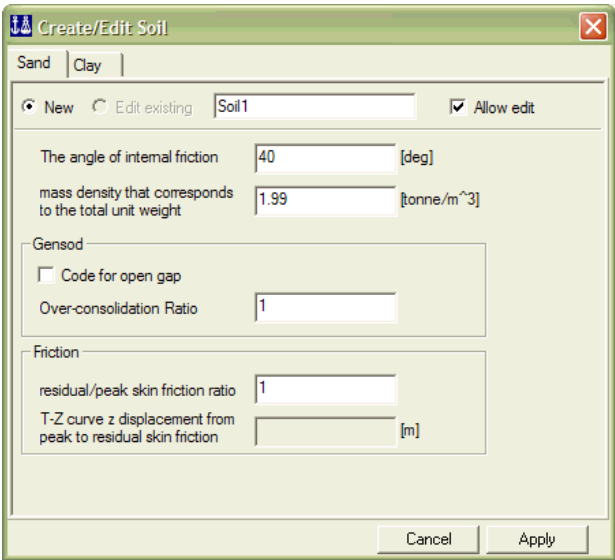


Figure 3.34 The Create/Edit Soil > Sand / Clay dialogs



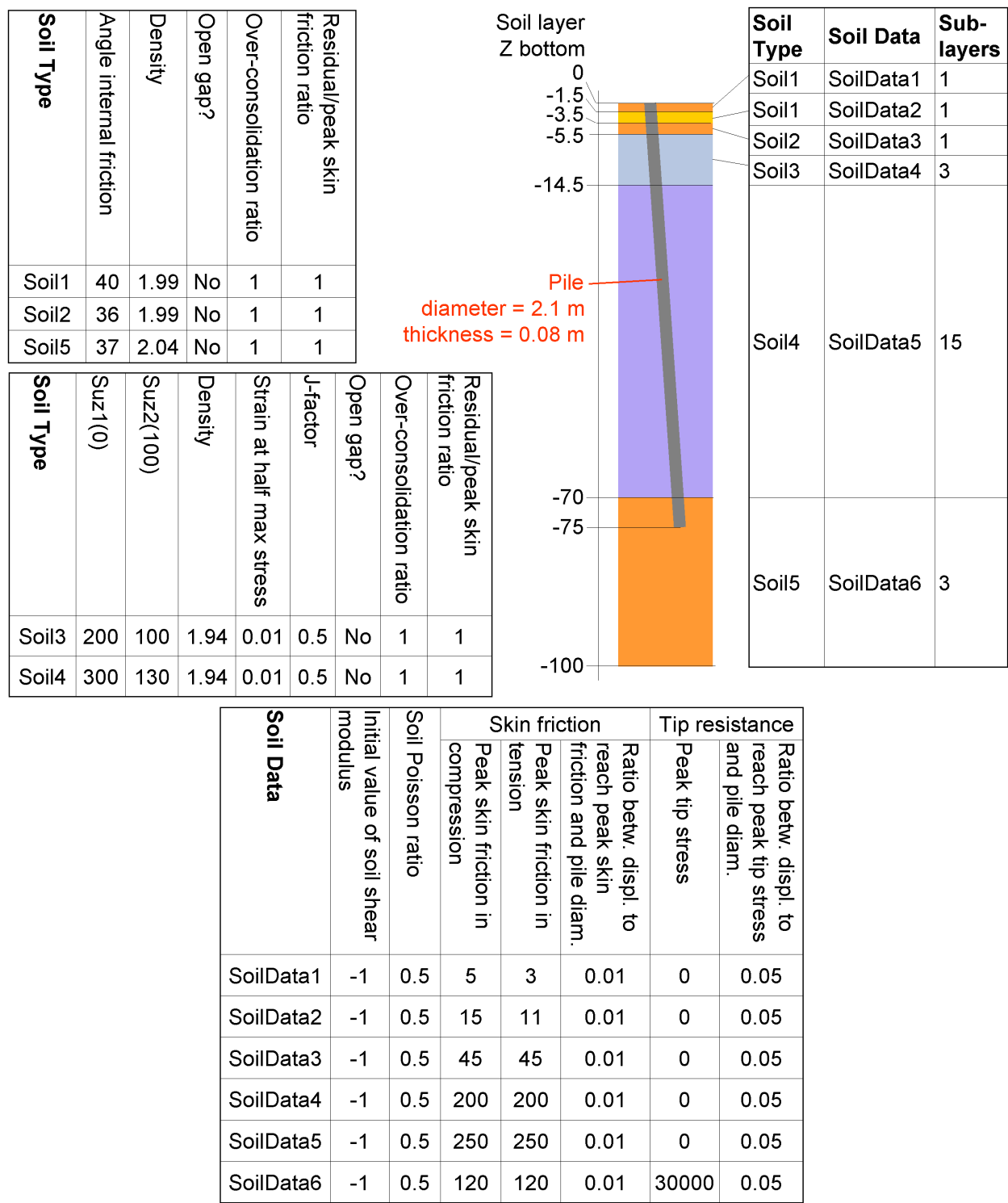


Figure 3.35 Sand and Clay data

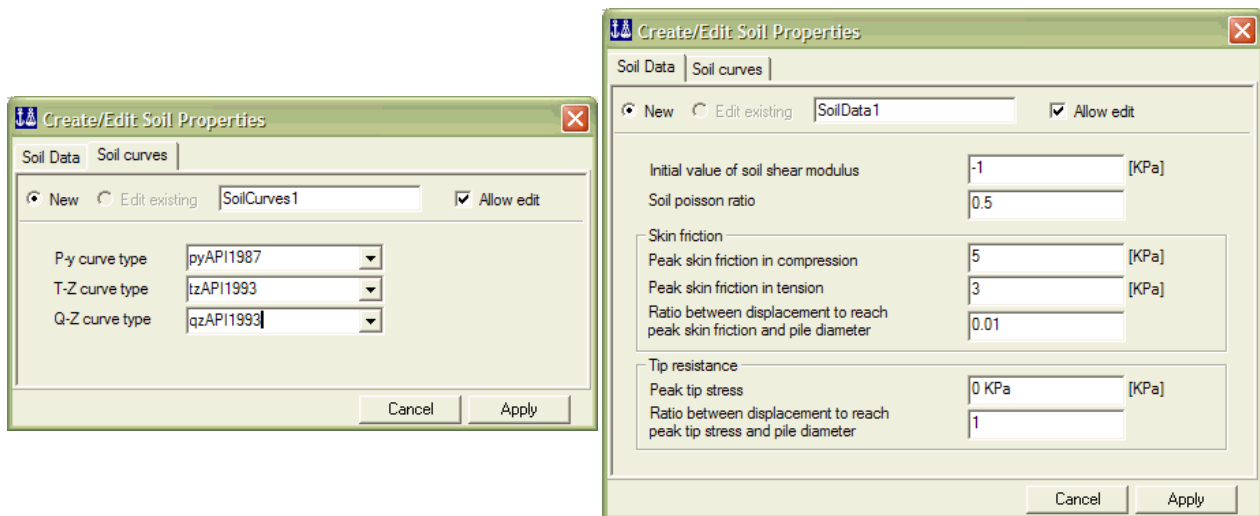


Figure 3.36 The Create/Edit Soil Properties dialog for soil curves and soil data

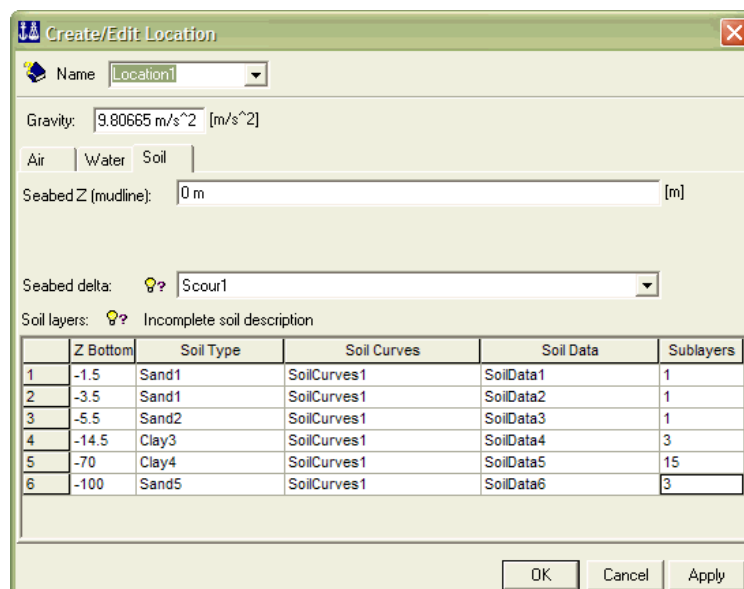


Figure 3.37 Fill in the Create/Edit Location dialog

The number of sublayers in the Create/Edit Location dialog controls the meshing of the piles, i.e. one node per sublayer. Each node is located in the middle of the sublayer.

Figure 3.38 shows the result of modelling the location with soil layers.

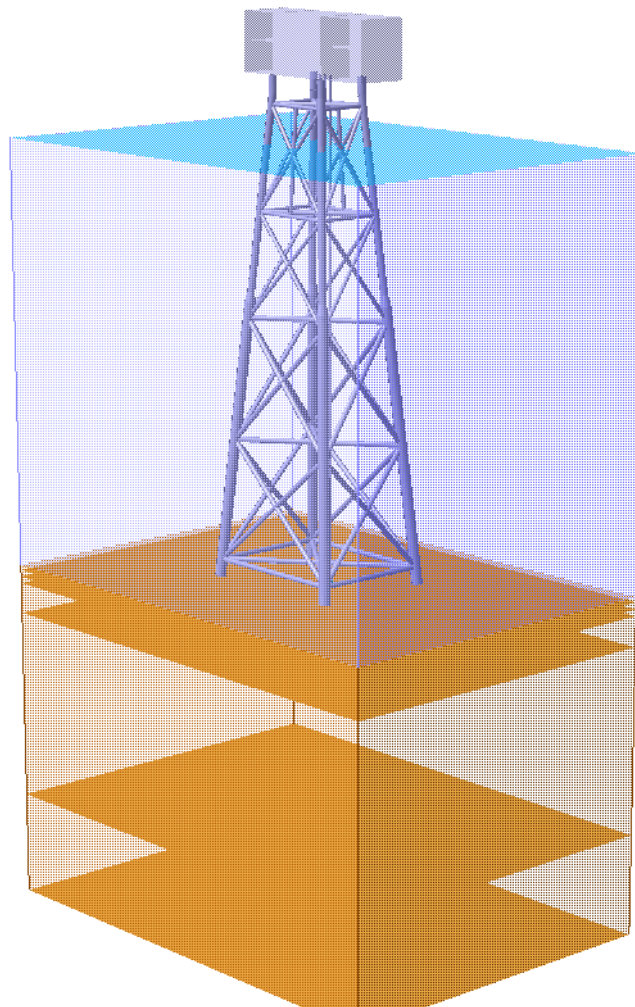


Figure 3.38 Location with soil layers (no piles yet)

### 3.6.4 Pile Modelling

First switch the Straight Beam button into a Pile button by clicking the down arrow next to the button. When the Pile button is pressed you are ready to insert piles. See Figure 3.39.

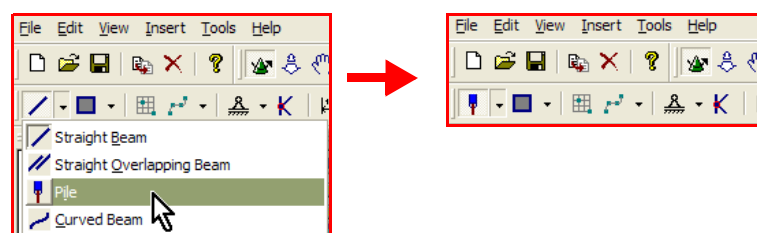
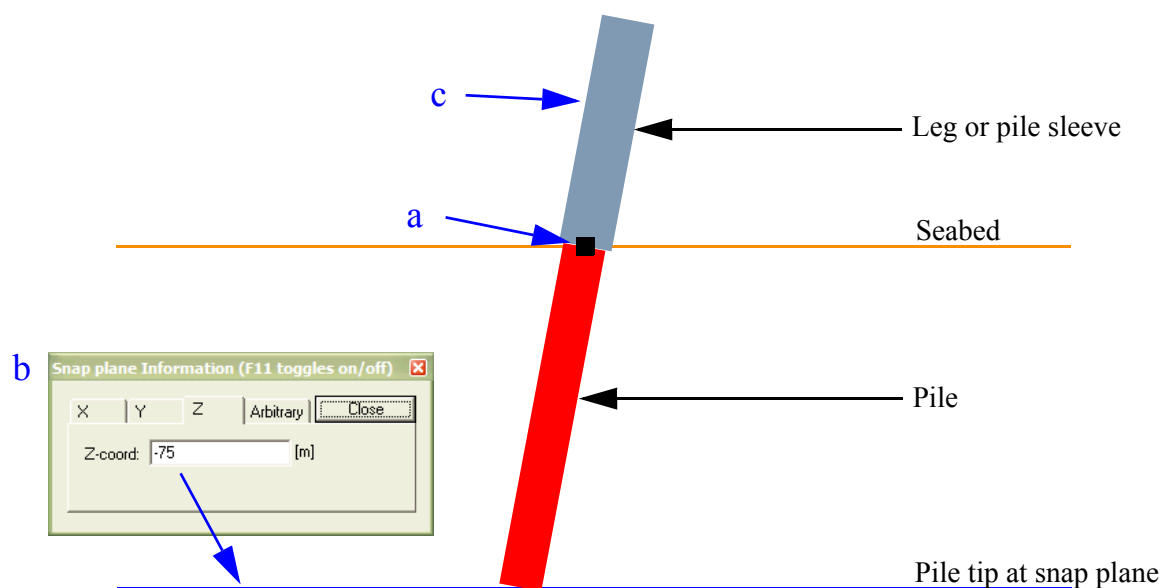


Figure 3.39 Create pile

A pile is easily modelled by the following three step procedure (the letters refer to Figure 3.40):

- a Select the pile top (click a snap point in the structure, e.g. the lower end of a leg or pile sleeve)
- b Specify the Z-level of the pile tip in the Snap plane dialog opening up (do not click anywhere in the model)
- c Align the pile by clicking an alignment beam in the structure, e.g. a leg or pile sleeve



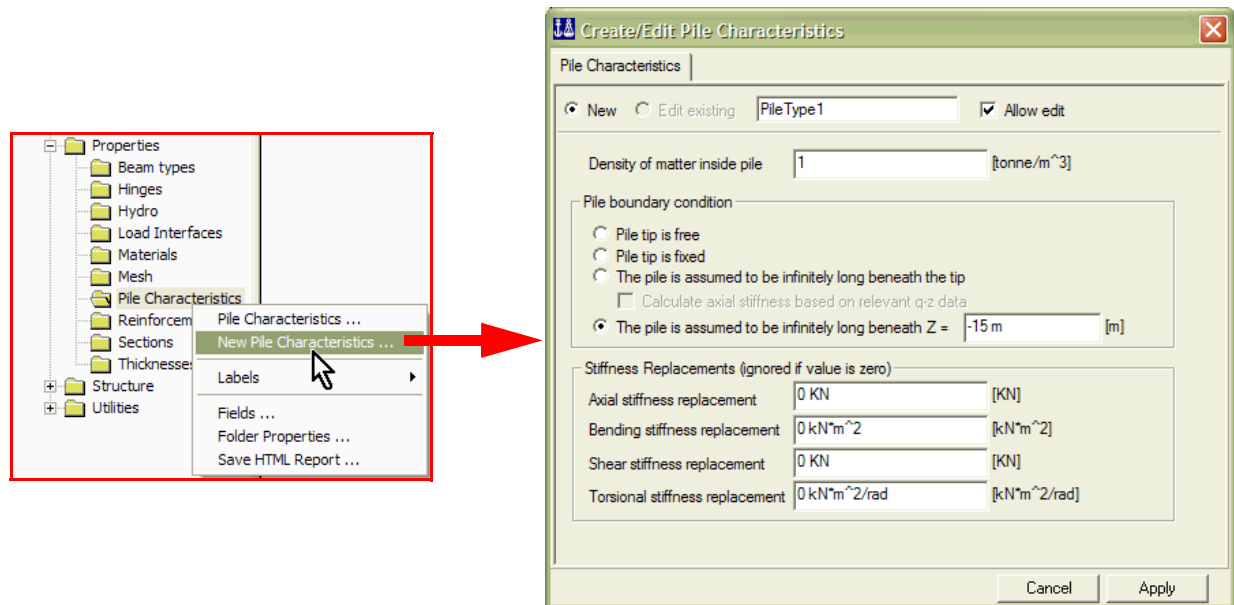
**Figure 3.40 Pile in pile sleeve**

The example jacket has four piles, one for each leg and as extensions of these. The following properties are required for the piles:

- Material, in the example use material Steel (a material created as part of the jacket structure input)
- Pipe section, in the example use section Pipe21 (a section created as part of the jacket structure input)
- Pile characteristics, see below

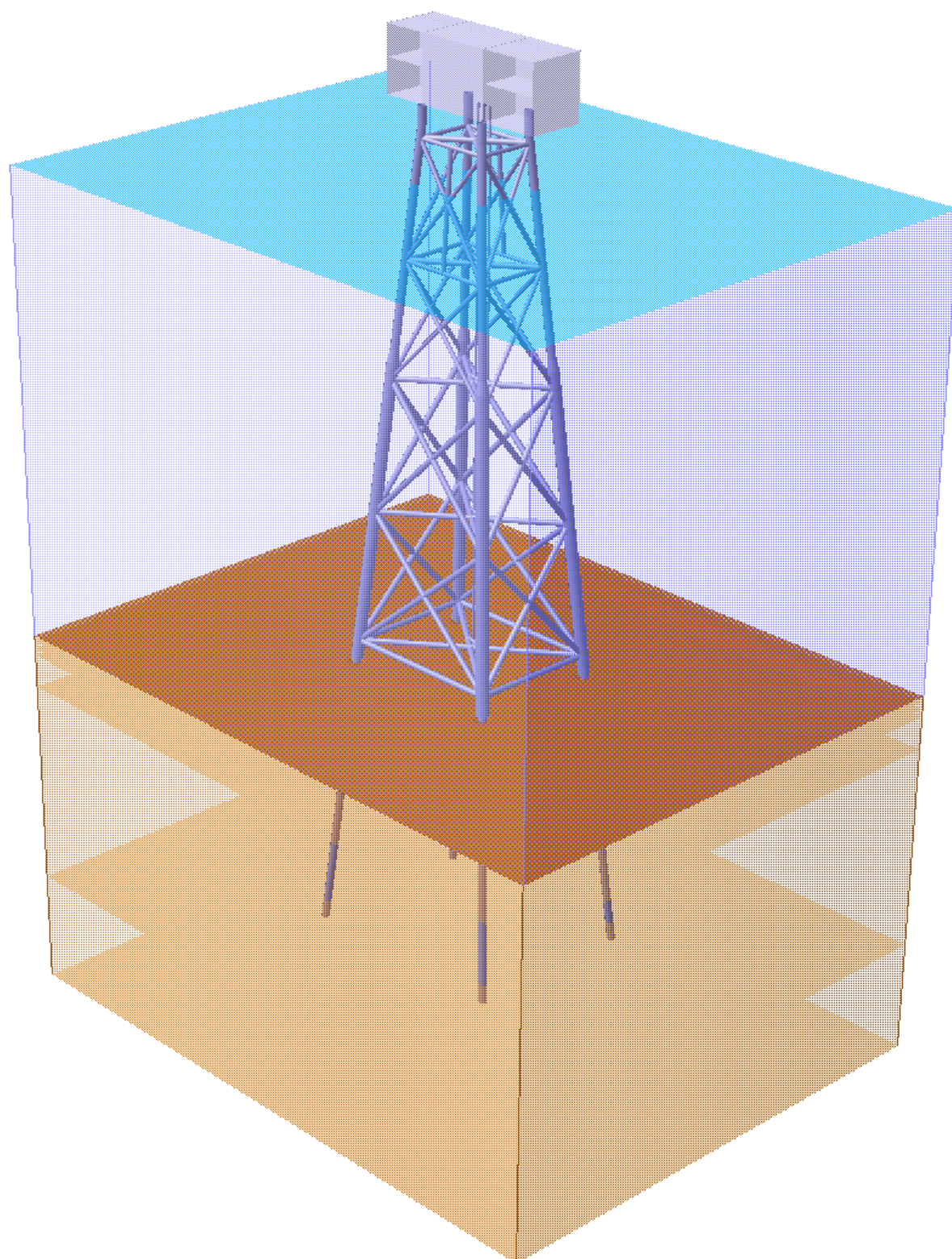
Assignment of material and section is done in the same way as for beams, i.e. either by setting defaults prior to modelling or by assigning the properties after modelling.

The Pile characteristics is a separate type of property allowing section stiffness values to be overruled (as compared to ungrouted Pipe section properties) and where special pile tip boundary conditions may be specified. Create the property by right-clicking the Properties > Pile Characteristic folder and selecting New Pile Characteristics; see Figure 3.41. Thereafter assign this property to all piles.



**Figure 3.41** The Create/Edit Pile Characteristics dialog

The complete jacket structure in the water with piles and soil environment is shown in Figure 3.42.



**Figure 3.42** The jacket in the water with piles and soil environment



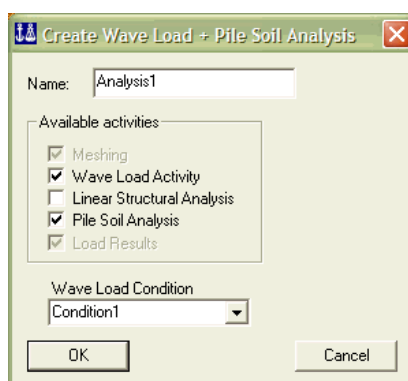
### 3.6.5 Define the Pile-Soil Analysis

Do an integrated wave load and structure-pile-soil analysis by defining a wave load condition. See Section 3.2 for details on how to work with wave load conditions and wave load analyses. In this example a regular wave set named WaveSet1 has already been defined. This regular wave set is selected in the New Wave Load Condition dialog. Figure 3.43 shows the table in the New Wave Load Condition dialog after selecting the regular wave set and adding current profile and wave model. A calm sea condition has also been added.

	Period	Height	Phase	Direction	Current profile	Wind profile	Wave model	Order
1	14 s	26 m	-60 deg	270 deg	CurrentProfile1		Stokes5	
2	12 s	15 m	-60 deg	180 deg	CurrentProfile1		Stokes5	
3	14 s	28 m	-60 deg	90 deg	CurrentProfile1		Stokes5	
4	15 s	27 m	-60 deg	0 deg	CurrentProfile1		Stokes5	
5							CalmSea	

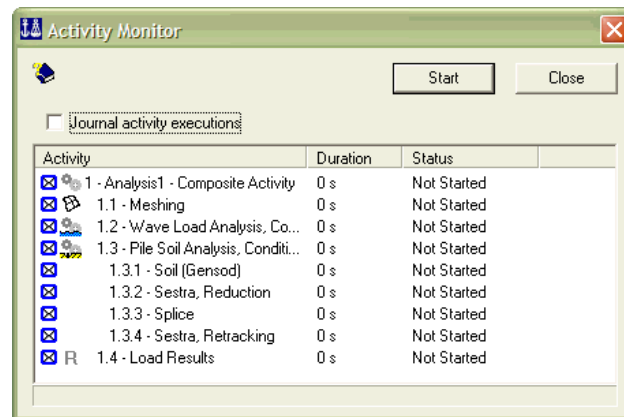
**Figure 3.43 Wave set from New Wave Load Condition dialog**

Having defined the wave loads the next step is to set up the analysis activity. This is done by right-clicking the Analysis > Activities folder and selecting New Analysis (or Alt+D). For a structure-pile-soil analysis including wave load calculation check the options shown in Figure 3.44 and specify a wave load condition.



**Figure 3.44 New analysis activity**

The activity is shown in the browser. Right-click the activity (or Alt+D) to open the Activity Monitor and view activity details; see Figure 3.45. This dialog is also used for activity execution.



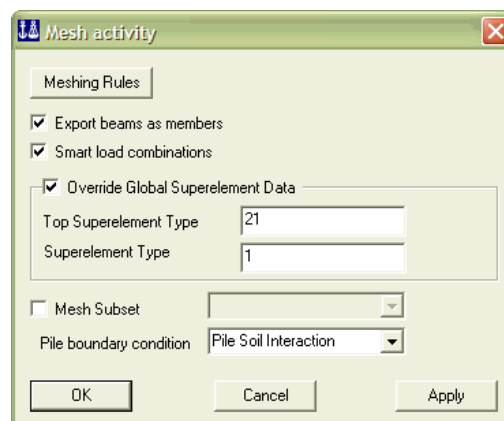
**Figure 3.45** The Activity Monitor dialog

**Note:** Right-click > Edit activity on the Meshing activity allows switching on ‘Smart load combinations’ (see the GeniE User Manual Vol. 1 about this). For a structure-pile-soil analysis Smart load combinations is irrelevant, i.e. this setting has no effect. (This is due to use of the superelement technique for such an analysis.)

### Using The Activity Monitor to Modify Analysis Control Parameters

Before executing the analysis by the Start button of the Activity Monitor parameters required by the solver programs must be set. These include:

- Meshing: To control whether and how piles are to be meshed and to set superelement numbers, in this case only set the top superelement number:
  - Right click > Edit activity to open the Meshing activity dialog and set top superelement number to 21; see Figure 3.46.



**Figure 3.46** The Mesh activity dialog for setting superelement number



- Wave load analysis: Control parameters for Wajac including defining the wave load cases and combinations:
  - Edit the wave load analysis as described in general in Section 3.2.3 and specifically in Figure 3.14.
  - Create load combinations as described in Section 3.2.4.
- Structure-pile-soil analysis: Control parameters for Sestra, Gensod and Splice, no action is required

For more details see the dialog reference in Section 5.6 and the Splice documentation.

### Scaling down Basic Load Cases

GeniE always analyses all loads, basic load cases as well as load combinations. However, in a non-linear pile-soil analysis only results for the load combinations are normally of interest as the non-linearity prevents results for the basic load cases to be combined.

The basic load cases are not only of little interest, in certain cases they may even cause the whole analysis to stop. For example, a wave load case alone (not combined with gravity) may cause tension in a pile. Depending on the pile and soil data such tension may cause the iteration of the non-linear analysis of Splice to fail and stop the complete analysis.

To circumvent this problem a feature involving scaling down the basic load cases is available in the form of a command. Click the Command Line area and type the command:

```
GenieRules.Meshing.basicLCfactor = 0.01;
```

### 3.6.6 Executing the Pile-Soil Analysis

Once the model has been created and analysis control parameters have been modified the complete analysis is run via the Activity Monitor. This includes the sub-activities:

- Meshing – creating a finite element representation
- Wave Load Analysis
- Pile Soil Analysis – the structure-pile-soil analysis that includes sub-sub-activities:
  - Soil (Gensod) calculation
  - Sestra, Reduction
  - Splice – the non-linear pile-soil interaction analysis
  - Sestra, Retracking
- Load Results back into GeniE for graphical presentation and reporting

Start the analysis by clicking Start and abort it at any time by clicking Abort; see Figure 3.47.

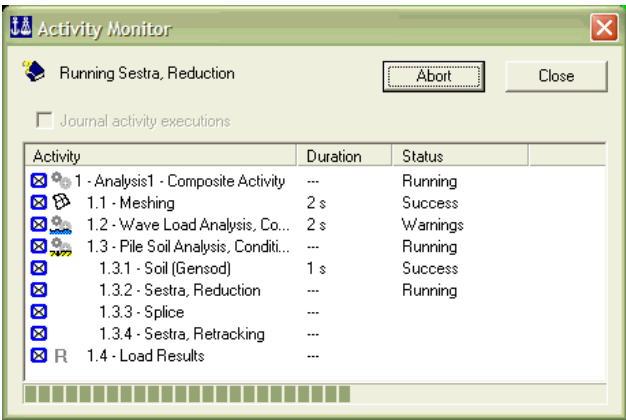


Figure 3.47 Execution via the Activity Monitor dialog

3.6.7 Reviewing Analysis Results

During execution the analysis programs Wajac, Sestra, Gensod, Splice or GeniE itself may report warnings or errors.

The Activity Monitor will indicate the status of each sub-activity in the Status column. Investigate details of a sub-activity by opening its listing file. This is done by right-clicking the sub-activity in the activity monitor; see Figure 3.48. The file selected will be opened using the editor associated with the particular file type. Search for the words ‘error’ and ‘warning’ in the listing file to get more information on the execution status.

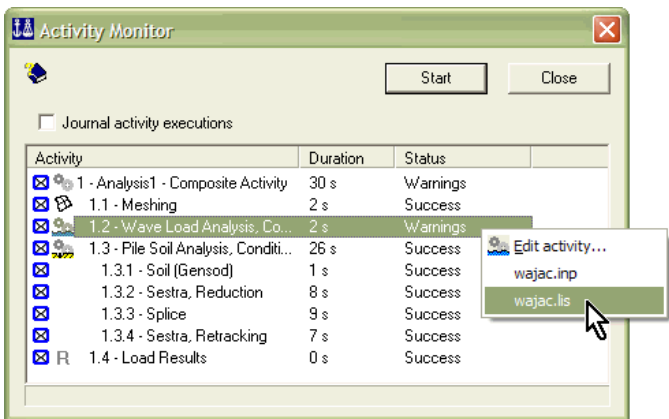


Figure 3.48 Opening listing file in an editor

After a successful execution the analysis results may be presented using standard results presentation features like deformed shape, bending moment diagrams, etc. Such result presentation may also be done for the piles.

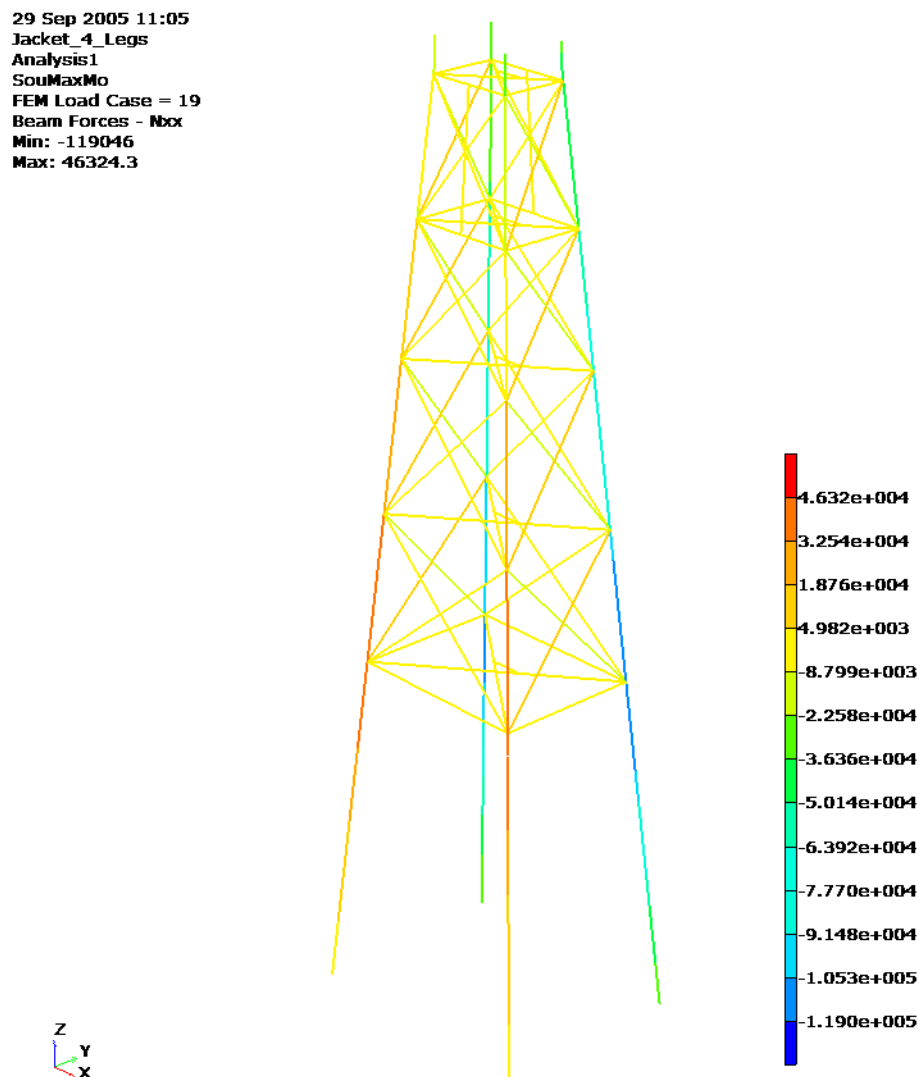


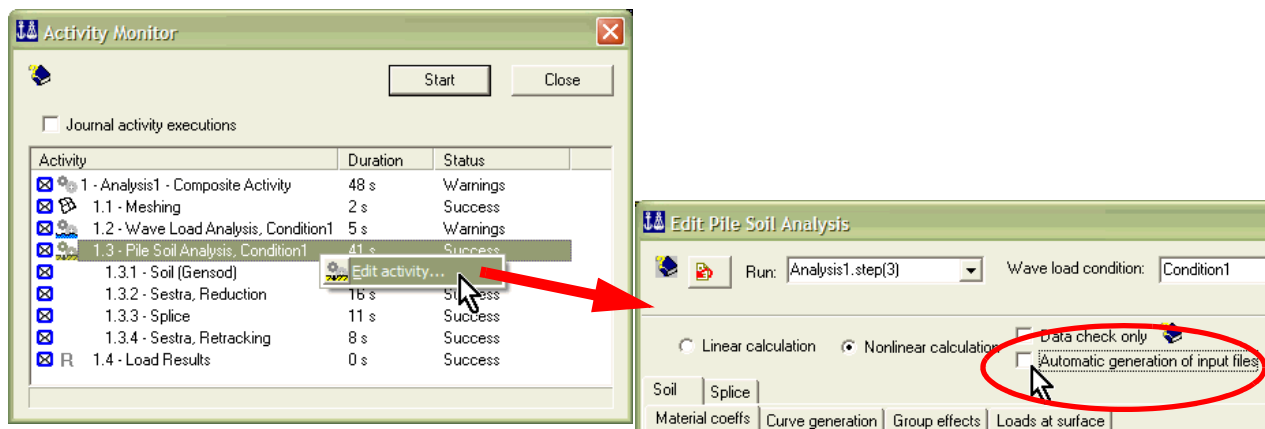
Figure 3.49 Axial beam force results for a piled jacket analysis

You may then modify the model or analysis control parameters and re-run the analysis.

### 3.6.8 Execution of the Pile-Soil Analysis with Modified Input

You may want to modify the pile-soil analysis (Gensod/Splice) input. This may be required in order to give input outside the capabilities of GeniE.

Do this by first right-clicking the activity '1.3 - Pile Soil Analysis, Condition1', selecting Edit activity, unchecking the Automatic generation of input files check box and clicking OK in the dialog; see Figure 3.50.



**Figure 3.50 Uncheck Automatic generation of input files in order to manually edit input**

Thereafter, right-click the appropriate activity, '1.3.1 - Soil (Gensod)' or '1.3.3 - Splice', and select gensod.inp/splice.inp – the input files to Gensod/Splice. The file appears in the text editor associated with the file type 'inp'. Edit the input as required and save the new version of the file using the same file name.

Click the Start button in the Activity monitor to start the revised execution.

**Note:** As long as the Automatic generation of input files box is unchecked changes to soil, pile and other data controlled by GeniE will not take effect in an analysis. When such changes are made in GeniE remember to check the Automatic generation of input files box. In turn, you may then need to uncheck the Automatic generation of input files box and redo the manual changes to the Gensod/Splice input file.

### 3.7 Doing the Analysis Outside GeniE

There may be cases when doing the wave load and structural analyses outside GeniE is required. Such cases are:

- When doing a superelement analysis
- To circumvent limitations in GeniE

In either case you need to export the FEM-file from GeniE by Tools > Analysis > Export FEM File. The exported file includes the whole model including piles, if present, and is identical to the FEM file automatically created during a wave load calculation or structure-pile-soil analysis controlled by GeniE. This FEM-file plus the input files to Wajac (wajac.inp), Gensod (gensod.inp) and Splice (splice.inp), if relevant, need to be copied to the project folder created by SESAM Manager.

Having exported the FEM-file and performed analyses controlled by Manager you cannot revert to GeniE for results presentation. In such cases use Xtract for general postprocessing and Framework for frame post-processing, both controlled by Manager.

## 4 EXECUTION OF GENIE: WAVE AND STRUCTURE-PILE-SOIL ANALYSIS

### 4.1 Program Environment

See the GeniE User Manual Vol. 1 for information on the hardware environment of GeniE and how to execute the program.

### 4.2 Files used by GeniE

GeniE (and Sestra, Wajac and Splice when started from GeniE) produces and uses the following files:

- GeniE journal file in Java script format – \*.js  
In addition to journalling (command logging) this file format is used for:
  - Reading a command file
  - Exporting a ‘clean’ js-file (File > Export > Genie journal file)
- GeniE database – \*.gni
- GeniE saved report in alternative formats – \*.lis, \*.html, \*.xml
- GeniE saved graphics in alternative formats – \*.gif, \*.eps, \*.bmp, \*.tga, \*.tif, \*.ps, \*.vrml, \*.dfile
- Genie\_log.txt file in case of model errors and/or warnings
- SESAM Input Interface File (neutral model file), also termed T-file – T\*.FEM
- SESAM Loads Interface File (produced by Wajac and a subset of the Input Interface File) – L\*.FEM
- SESAM Matrix Interface File (data exchange file between Sestra and Splice) – M\*.SIF
- SESAM Results Interface File (neutral model plus results file) – R\*.SIN (or R\*.SIF or R\*.SIU)

- Batch program input (input to Wajac, Gensod, Splice, Sestra) – \*.inp
- Batch program listing (print) file (from Wajac, Sestra, Gensod, Splice) – \*.lis
- Simulation Result File (from Wajac) – \*.ser (See the Wajac User Manual, the LOASIM command, for details.)
- Batch program message file – \*.mlg
- Sestra maintenance file – \*.mnt

## 5 DIALOG DESCRIPTION

### 5.1 Dialog Features

This section describes useful features common for several of the dialogs explained below. The subsequent three sections (Section 5.2, Section 5.3 and Section 5.4) present all dialogs relevant when doing a wave load analysis. Additional information on data and special features for each dialog is given. The descriptions are grouped in separate chapters for the Environment, the Hydro Properties and the Wave Load Analysis.

#### 5.1.1 Dialog Help

Help information is available in the dialogs and is activated by positioning the cursor over the help icons:



A dialog contains at the most one such book icon. It normally gives a general description of the dialog and its purpose; see Figure 5.1.

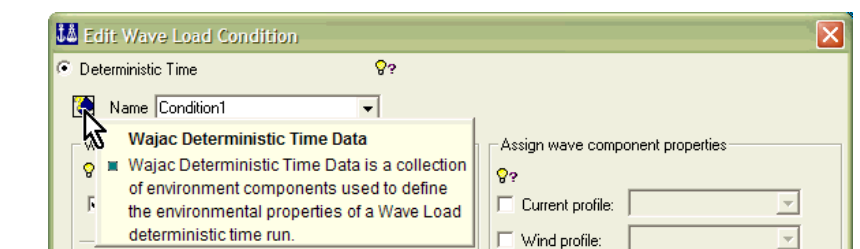


Figure 5.1 Example of use of book icon



A dialog contains any number of such icons. Normally, each is positioned nearby a data field or a feature for which it gives an explanation; see Figure 5.2.

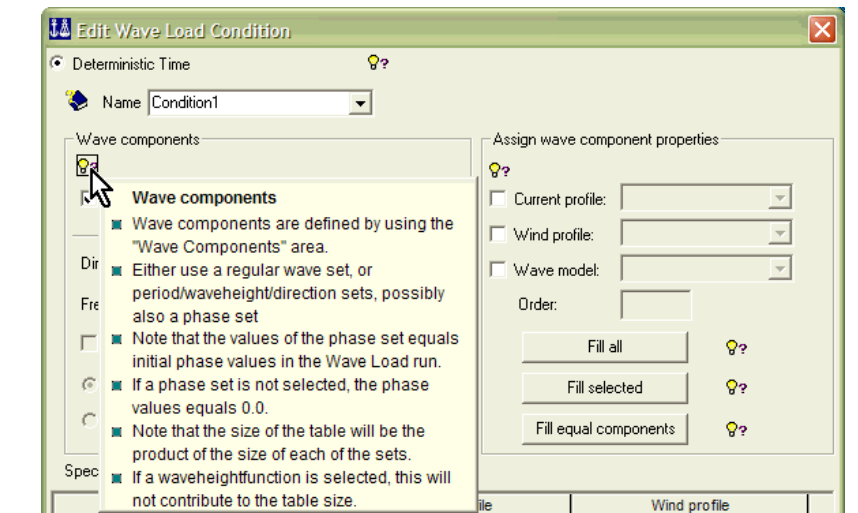


Figure 5.2 Example of use of light bulb icon

### 5.1.2 Dialog New/Edit Functionality

Many dialogs have a special functionality for handling both creation of a new data object and editing an existing. When the dialog is opened in 'new-mode' the New and Allow edit options are automatically ticked and the dialog is ready to accept data. When opened in 'edit-mode' the Edit existing option is ticked. All data fields are disabled and the dialog is in a read-only state. To open for editing the Allow edit option must be ticked.

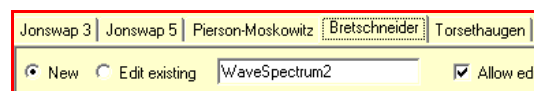


Figure 5.3 New/edit functionality in dialogs

### 5.1.3 Late Definition Functionality

Some dialogs offer what may be called 'late definition functionality'. Such functionality is found in dialogs that include other data objects. An example is the Wave Load Condition dialog (Figure 5.36) where dialogs for defining waveset, current and wind profile objects may be activated directly from this dialog. For example, the regular waveset dialog is activated from the Regular waveset combobox in the Wave components area by selecting <Create new RegularWaveset>. Thus, no data need to be defined up front.

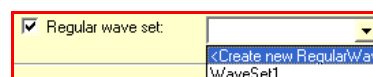


Figure 5.4 Late definition functionality


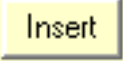




### 5.1.4 Table Functionality

Dialogs that contain a table will often have the functionality described below. This functionality makes it easier to enter and modify table value. In addition to writing values directly into the table the following options exist:

- Copy from a spreadsheet

Values in a spreadsheet, e.g. Excel, may be copied and pasted into the dialog provided that the number of columns is the same. The number of rows in the table of the dialog must, if necessary, be increased (position the cursor in the last row and click the down arrow) to allow the rows of the spreadsheet to be copied.

-  If the values have a systematic variation Fill table may be used. Enter first, last and step values and click the Fill table button. Values are automatically filled in the table.
-  The Insert button is used to insert one or more values. Number of values inserted equals number of rows selected. Values are inserted before first selected row and are calculated based on linear interpolation.
-  The Remove button removes all selected rows.
-  The Clear table button clears the table.

Some common table features:

- To select a row click to the left of the row. The complete row is highlighted.
- Add more rows by positioning the cursor in the last row and clicking the down arrow.
- By positioning the mouse over the heading (above the first row) useful hints on data to enter are provided; see Figure 5.5.

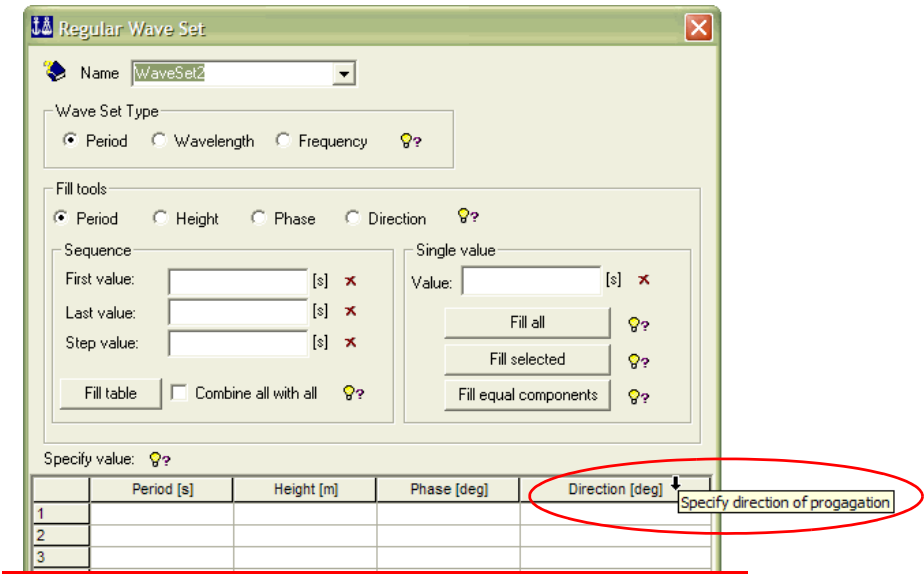


Figure 5.5 Table head hint


## 5.2 The Environment

The environment data describes the environmental part of an analysis, e.g. wind, current, waves and sea depth in addition to constants like gravity and water density. The data consists of the following main data types: Location, Directions, Air, Water and Soil. Soil is, however, not relevant for wave load analysis.

### 5.2.1 Location

A location defines the environmental features at the site of the structure. It is natural to define a location as the first task when building the environment model.

The Location dialog appears with modifiable default values. It includes three tabs: Air, Water and Soil. Values in the Air tab normally remain unchanged. The Waterline Z value in the Water tab and the Seabed Z value in the Soil tabs normally need to be modified. These values are given in the model (global) coordinate system.

When the location is created the layers and borders appear in the graphical window, as shown in Figure 3.2. Colours, transparencies and whether an object is visible or not are easily modified by the Location selection button . By default the air layer is invisible as this often extends far above the structure. By using the Location selection button functionality the visibility is easily turned off. When selecting parts of the structure the water layer might also be removed. Note, however, that the 'Modelling - Structure' and 'Modelling - Transparent' display configurations by default do not include the Location (air, water and soil). Read about changing and making new display configurations (model views) through the View Options dialog in the GeniE User Manual Vol. 1, Section 3.1.

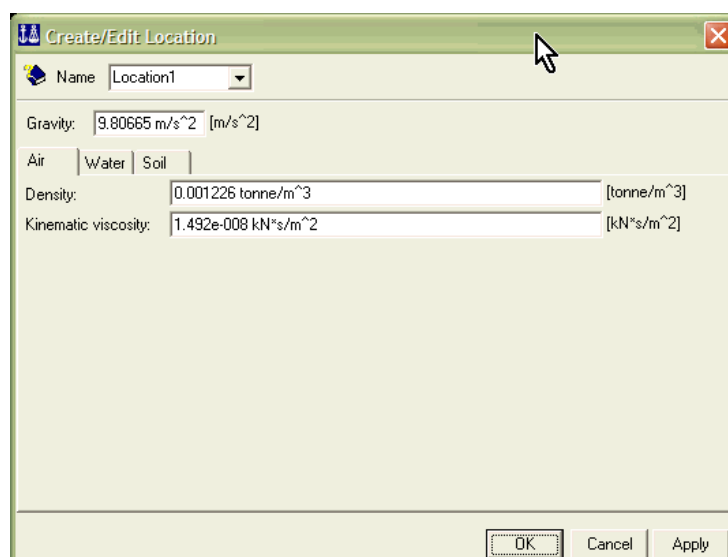


Figure 5.6 The Create/Edit Location > Air dialog

## Soil

Seabed Z is the unscored mudline given in the global coordinate system.

Seabed delta is a property used for defining scour.

Z Bottom of each layer is given in the global coordinate system (NOT as distance to mudline). The layers must be entered in order of decreasing Z.

Soil Type is a property with soil type and principal characteristics.

Soil Curves is a property which controls the generation of P-Y/T-Z/Q-Z curves

Soil Data is property with additional soil characteristics.

Sublayers is the number of subdivisions of each layer. The number of sublayers controls the number of nodes along the pile. Enter values so that you get a fine mesh near the soil surface.

**Create/Edit Location**

Name: Location1

Gravity: 9.80665 m/s<sup>2</sup> [m/s<sup>2</sup>]

Air | Water | **Soil**

Seabed Z (mudline): 0 m [m]

Seabed delta:

Soil layers:

	Z Bottom	Soil Type	Soil Curves	Soil Data	Sublay
1					
2					
3					
4					
5					

OK Cancel Apply

**Figure 5.7 The Create/Edit Location > Soil dialog**

## 5.2.2 Directions

There are two types of direction data: a single direction and a set of directions. Only the set is relevant for wave load analysis.

Direction sets are used for defining a range of wave propagation directions. Specify angles between positive x-axis and direction of wave propagation.

In a wave load analysis it is optional. It may be used for defining a Wave Load Condition either of type Deterministic Time (Figure 5.17) or Frequency Domain (frequency domain is not available in the 3.2 version of GeniE).

The screenshot shows the 'Create/Edit Direction' dialog box with the 'Direction Set' tab selected. The 'Name' field is set to 'DirectionSet1'. The 'First direction' is 0 [deg], 'Last direction' is 360 [deg], and 'Step' is 15 [deg]. There are buttons for 'Fill table', 'Row operations', 'Specify direction', 'Insert', 'Remove', and 'Clear table'. At the bottom, there is a table with 10 rows and a column header 'Direction [deg]'. The 'Cancel' and 'Apply' buttons are at the bottom right.

	Direction [deg]
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Figure 5.8 The Create/Edit Direction > Direction Set dialog

### 5.2.3 Air

Wind profile is the only air data type. It must be defined if wind loads are to be calculated.

Three different wind profile formulae may be selected: General, Normal and Extreme. They correspond to equations 2.27-2.29 in the Wajac User Manual.

The Direction relative to x-axis option is default. However, by using the Direction relative to wave heading option and setting Wind direction to 0 degrees the wind is aligned with the wave heading. In many cases it will then be sufficient to define a single wind profile.

In a wave load analysis it may be used when defining a Wave Load Condition of type Deterministic Time (Figure 5.17).

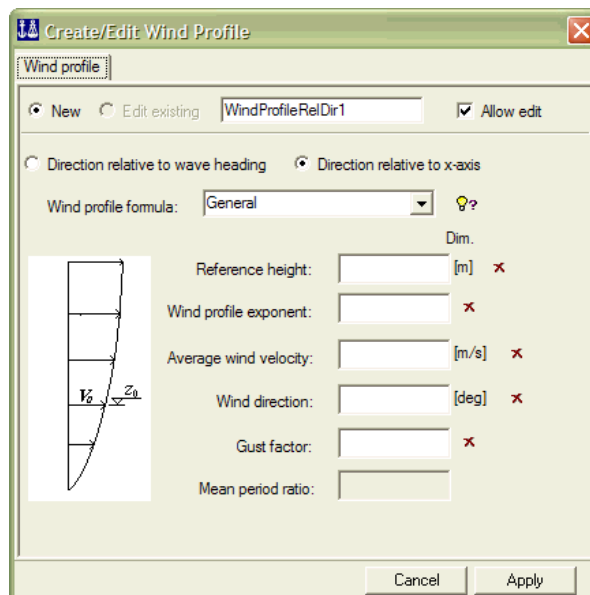


Figure 5.9 The Create/Edit Wind Profile dialog

### 5.2.4 Water

The following types of water data are relevant for wave load analysis: current profile, frequency (period) set, phase set, wave height set, wave height function, regular wave set and wave spectra (Jonswap 3 parameter, Jonswap 5 parameter, Bretschneider and Pierson-Moskowitz).

#### Current Profile

In the dialog the Direction along wave heading option is default. However, by using the Direction along wave heading option the current is aligned with the wave heading. In many cases it will then be sufficient to define a single current profile.

The elevations must be given in sequence of decreasing Z-values and the Z-value is always 0 at the sea surface and decreasing downwards.

In a wave load analysis a current profile may be used when defining a Wave Load Condition (Figure 5.17).

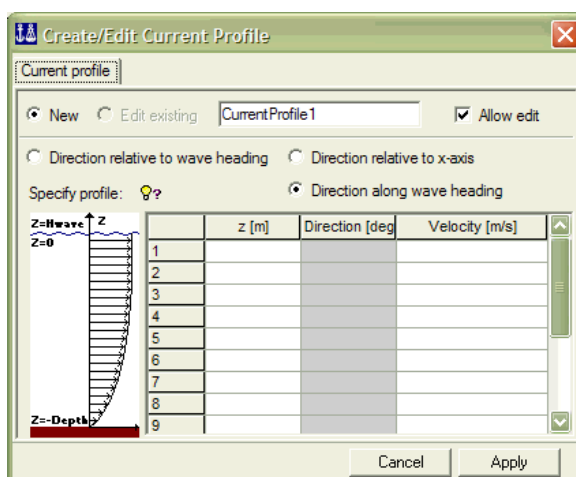


Figure 5.10 The Create/Edit Current Profile dialog

Frequency Set

The dialog has three options: Period, Wavelength and Frequency. The Wavelength option is illegal in this context. For a wave load analysis Period is the normal choice. However, if Frequency data is used it will be converted to Period in the Edit Wave Load Run dialog (Figure 5.17).

In a wave load analysis a frequency set may be used when defining a Wave Load Condition (Figure 5.17).

**Note:** Frequency domain is not available in the 3.2 version of GeniE.

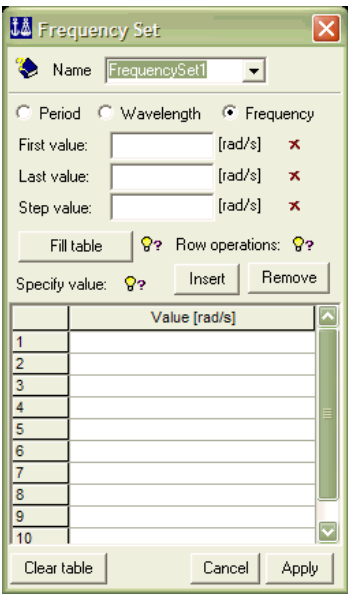


Figure 5.11 The Frequency Set dialog



## Phase Angles Set

The phase angles will serve as initial phase angles in the seastates defined in the Edit Wave Load Run dialog (Figure 5.43).

In the Wave Load Condition dialog (Figure 5.17) the default behaviour is that no phase angle set is selected in which case the initial phase angle is 0.0 degrees. If the initial phase angle should have another value or more values then a set must be defined and selected.

In a wave load analysis a phase angle set may be used when defining a Wave Load Condition.

Phase Angles Set

Name: PhaseSet1

First angle: 0 [deg]  
Last angle: 345 [deg]  
Angle step: 15 [deg]

Fill table ? Row operations: ?  
Specify phase: ? Insert Remove

	Phase [deg]
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Clear table Cancel Apply

Figure 5.12 The Phase Angles Set dialog

Wave Height Set

In a wave load analysis a wave height set may be used when defining a Wave Load Condition (Figure 5.17).

Create/Edit Wave Height

Wave Height Function | Wave Height Surface | Wave Height Set

Name

WaveHeightSet1

First height:

[m]

✖

Last height:

[m]

✖

Step height:

[m]

✖

Fill table

?

Row operations:

?

Specify value:

?

Insert

Remove

Clear table

	Value [m]
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Cancel

Apply

Figure 5.13 The Create/Edit Wave Height > Wave Height Set dialog

## Wave Height Function

In a wave load analysis a wave height function may be used when defining a Wave Load Condition (Figure 5.17).

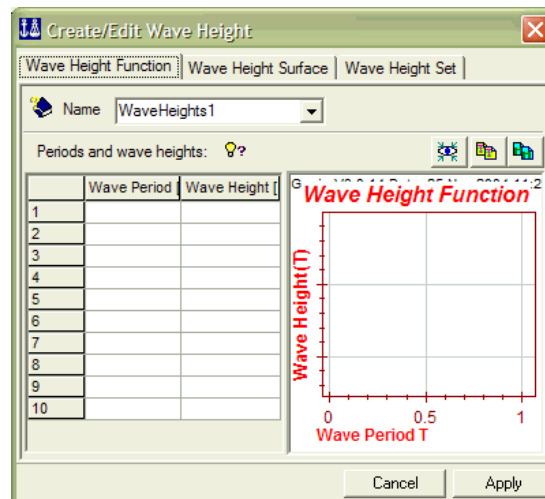


Figure 5.14 The Create/Edit Wave Height > Wave Height Function dialog

## Regular Wave Set

When defining a regular wave set neither Direction set, Frequency set, Phase set nor Wave height set/function are needed. For such cases there is a simple and fast way of producing input for creating a Wave Load Condition. With the Regular Wave Set dialog the user defines an arbitrary number of wave components. Each wave must contain values for period, height, phase angles and direction. The phase angles will serve as initial phase angles in the seastates defined in the Edit Wave Load Run dialog (Figure 5.43). The dialog contains separate areas to help the user enter values. In particular, the wave components may either be entered manually into the table or the Fill tools may be used.

**Figure 5.15 The Regular Wave Set dialog**

Wave Set Type area:

- The user must specify whether Period or Frequency shall be used for the regular wave set.
- Wavelength is not allowed in wave load analysis.

Fill Tools area:


- Values are specified in the Sequence or Single value area.
- For each time a Fill button is clicked the table is updated.

- Phase values equals initial phase values in the wave load run.

Sequence area:

- Each time the Fill table button is clicked the sequence of values specified will be appended to the values already existing in the selected column.
- If Combine all with all is ticked then N times M rows are generated, N is the number of rows before the fill and M is the number of values in the sequence.

Single value area:

- All rows will be filled with the single value when the Fill all button is clicked.
- Selected rows will be filled with the single value when the Fill selected button is clicked.
- To use the Fill equal components button it is necessary to select a cell that contains a value that will be used for selecting relevant rows.
- For more information use the  icons.

A regular wave set may be used when defining a Wave Load Condition (Figure 5.17).

## Wave Spectrum

Four wave spectra are relevant in a wave load analysis: 3 and 5 parameter Township, Pierson-Moskowitz and Bretschneider. The dialogs are shown in Figure 5.16.

A wave spectrum may be used when defining a Wave Load Condition.

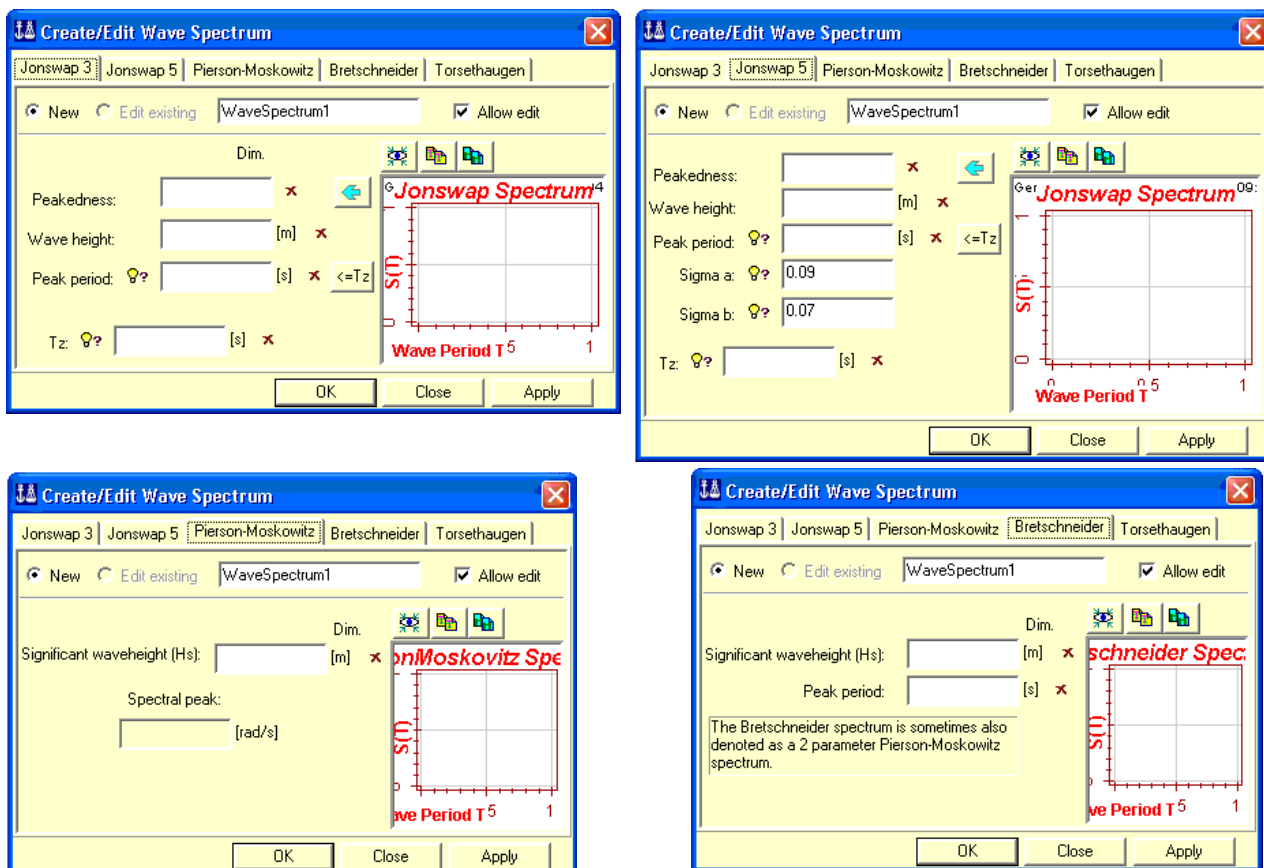


Figure 5.16 The Create/Edit Wave Spectrum dialogs

## Wave Load Condition

The Wave Load Condition dialog has two modes: Deterministic Time and Frequency Domain. This reflects what type of wave load calculation the condition will be used for:

- Deterministic time means that the data set will be used in a deterministic load calculation.
- Frequency domain means that the data set will be used to calculate transfer functions

**Note:** Frequency domain is not available in the 3.2 version of GeniE.

A Wave Load Condition will be used in the Edit Wave Load Run dialog (Figure 5.43).

### Deterministic time mode

The dialog offers a flexible way to define wave components with additional properties, i.e. environmental properties necessary for doing a Wave Load deterministic time run. Initially, the wave components are defined by using the Wave Components area. Thereafter, properties are added to each component either by using functions in the Assign wave component properties area or by manually entering properties by using comboboxes inside the table. The Period, Height, Phase and Direction columns are locked. This is done to force the user to update the original data objects and thus secure data consistency throughout the model.

The Wave components area is for defining wave component data. The relevant comboboxes will contain all existing objects of that type plus an option to define new objects.

- Either use a Regular Wave Set or a combination of Direction Set, Frequency (Period) Set, Phase Angle Set (optional) and Wave Height Set/Function.
- The values of the phase set are the initial phase values in the wave load run.
- If a phase set is not selected the phase values are 0.0.
- The size of the table will be the product of the size of each of the sets.
- If a Wave Height Function is selected this will not contribute to the table size. The actual wave periods are calculated by interpolation of the wave height function.

The Wave components with properties area is for properties that may be assigned to a wave component. Such properties are current profile, wind profile and wave model. The only mandatory property is wave model. The wave model combobox has the following options: Airy, CalmSea, Cnoidal, Stokes' 5th and Dean's stream function. In the last case the order must be given as an integer between 1 and 24.

- Properties may be assigned in several ways, and several properties can be filled at the same time.
- What properties to fill, current profile/wind profile/wave model, are controlled by the respective check boxes.
- If Stream function is selected for Wave model the order must be given as well.
- Each time a Fill button is clicked the table in the Wave components with properties area is updated.

- Calm sea is specified by appending a row with no values for period, height, phase and direction. The Wave model is specified as CalmSea.
- The calm sea row(s) must follow after the rows containing wave data.
- For an explanation of how the Fill all, Fill selected and Fill equal buttons function see the Regular Wave Set dialog (Figure 5.17).

**Note:** When CalmSea is the only Wave model specified, i.e. there are no waves and no current, then the red X indicating missing specification of Regular wave set should be neglected.

**New Wave Load Condition**

☒ Deterministic Time

Name: Condition1

**Wave components**

☒ Regular wave set: [ ]

Direction set: [ ]

Frequency set: [ ]

☐ Phase set: [ ]

☒ Wave height set: [ ]

☐ Wave height function: [ ]

**Assign wave component properties**

☐ Current profile: [ ]

☐ Wind profile: [ ]

☐ Wave model: [ ]

Order: [ ]

Buttons: Fill all, Fill selected, Fill equal components

Specify value: [ ]

	Period	Height	Phase	Direction	Current profile	Wind profile	Wave model	Order
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								

Location: Location1

Buttons: OK, Cancel, Apply

**Figure 5.17 The New Wave Load Condition dialog – Deterministic Time**

## Frequency domain mode

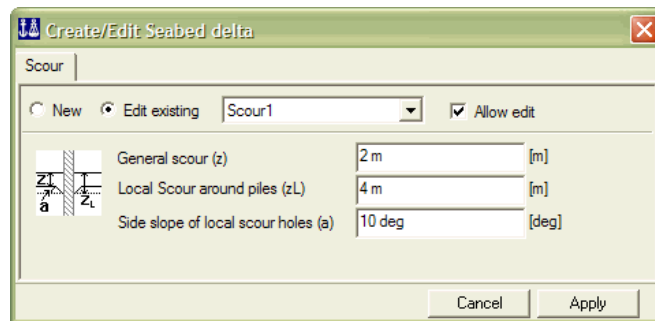
**Note:** Stochastic (spectral) fatigue wave analysis is not offered by the 3.2 version of GeniE.

Currently, such analysis is performed by exporting the FEM-file from GeniE by Tools > Analysis > Export FEM File (see Section 4.2 about files used by GeniE) and controlling the analysis from SESAM Manager. See Section 3.7 about how to do analyses outside GeniE.



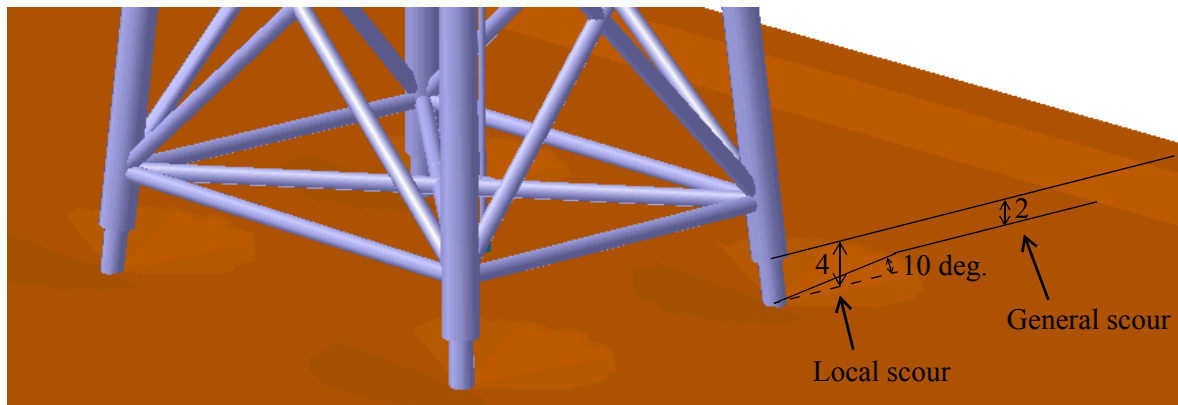
### 5.2.5 Soil

#### Scour



**Figure 5.18** The Create/Edit Seabed delta dialog

Two types of scour may be accounted for: general scour over a large area around the pile and a local conical scour hole around each pile; see Figure 5.19.



**Figure 5.19** General and local scour

These scour types will influence the vertical stresses down along the pile and hence the lateral resistance that can be mobilised.

- General scour is the depth of general scour relative to Seabed Z (mudline).
- Local scour is the depth to the bottom of the local scour holes around the piles. Local scour is also given relative to Seabed Z (mudline). The local scour must be equal to or greater than the general scour.
- Side slope of the local scour hole is given as an angle. A small angle gives a wide hole.

## Create Soil – Sand

**Figure 5.20 The Create/Edit Soil > Sand dialog**

- Angle of internal friction given in degrees
- Mass density corresponding to the total unit weight
- Code for open gap (if checked, otherwise closed gap is assumed) behaviour for cases where either reversed loading takes place or where an open hole is specified
- Over-consolidation ratio, i.e. ratio of maximum previous vertical stress to present vertical stress. The value is used by the DNV code to distinguish normally and overly consolidated clays.
- Skin friction:  
The two values are used to describe a reduction in skin friction against the pile after the peak value has been reached. The values are used in connection with T-Z codes 103, 200 and 293 only.
- Residual/peak skin friction ratio is the ratio between residual skin friction (at large strains) and peak skin friction.
- T-Z curve Z displacement from peak (skin friction) to residual skin friction

## Create Soil – Clay

The screenshot shows the 'Create/Edit Soil' dialog box with the 'Clay' tab selected. The 'New' radio button is chosen, and the soil name is 'Soil1'. The 'Allow edit' checkbox is checked. The 'Undrained Shear Strength' section has fields for 'suz1' [Pa], 'suz2' [Pa], 'Z1' [m] (0 m), and 'Z2' [m] (-100 m). The 'mass density that corresponds to the total unit weight' field is empty. The 'API' section has 'strain at half of max stress' empty and 'J factor for API code' set to 0.25. The 'Gensod' section has 'Code for open gap' unchecked and 'Over-consolidation Ratio' empty. The 'Friction' section has 'residual/peak skin friction ratio' set to 1 and 'T-Z curve z displacement from peak to residual skin friction' empty. The 'Cancel' and 'Apply' buttons are at the bottom right.

**Figure 5.21 The Create/Edit Soil > Clay dialog**

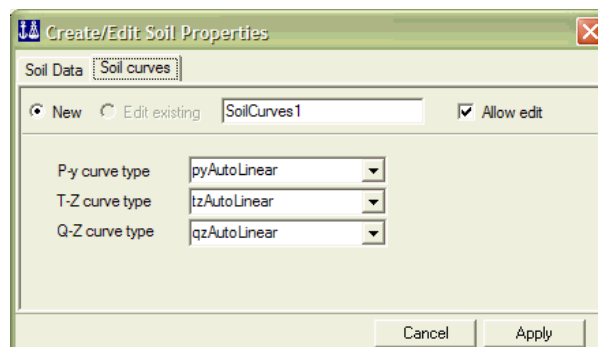
- Undrained shear strength at  $Z1 = 0$  and at  $Z2 = -100$  valid for the present soil type. At elevation  $Z$  the  $su$  value is given by:  

$$su(Z) = su(Z1) + (su(Z2) - su(Z1)) * Z / (Z2 - Z1)$$
- Mass density corresponding to the total unit weight
- Strain at half of max stress for laboratory undrained compression tests carried out on undisturbed soil samples. This value is required both by the API and the DNV codes.
- J factor for API – J factor required by the API code to find the depth of the reduced resistance zone. API recommends a value of 0.25 to 0.50.
- Code for open gap (if checked, otherwise closed gap is assumed) behaviour for cases where either reversed loading takes place or where an open hole is specified
- Over-consolidation ratio, i.e. ratio of maximum previous vertical stress to present vertical stress. The value is used by the DNV code to distinguish normally and overly consolidated clays.
- Skin friction:

The two values are used to describe a reduction in skin friction against the pile after the peak value has been reached. The values are used in connection with T-Z codes 103, 200 and 293 only.

- Residual/peak skin friction ratio i.e. Ratio between residual skin friction (at large strains) and peak skin friction.
- T-Z curve Z displacement from peak (skin friction) to residual skin friction

## Soil Curves



**Figure 5.22 The Create/Edit Soil Properties > Soil curves dialog**

- P-Y curve types
  - pyManual – Manual input of p and y values for each layer and pile diameter (p-y code 0).
  - pyAutoLinear Automatic generation – linear (p-y code 100).
  - pyAutoBilinear Automatic generation – bilinear (p-y code 101).
  - pyAutoTrilinear Automatic generation – trilinear (p-y code 102).
  - Automatic generation using a set of simple rules. The p max value is calculated from J. Brinch Hansen (1961) earth pressure theory, displacement y to reach p max is set to 0.1 times pile diameter. All p y curves pass through this point, the value of N determines curve shape.
  - pyAPI1984 – p-y data will be generated by the API 1984 edition rules (p-y code: 284).
  - pyAPI1987 – p-y data will be generated by the API 1987 edition rules (p-y code: 287).
  - pyDNV1980 – p-y data will be generated by the DNV 1980 edition rules (p-y code: 380).
  - pyISO2004 – p-y data will be generated by the ISO2004 draft rules (p-y code: 404).
- T-Z curve types
  - tzManual – Manual input of t and z values for each layer and pile diameter (t-z code 0).
  - tzAutoLinear Automatic generation – linear (t-z code 100).
  - tzAutoBilinear Automatic generation – bilinear (t-z code 101).
  - tzAutoTrilinear Automatic generation – trilinear (t-z code 102).
  - tzKraft – t-z data will be generated by the Kraft et al. (1981) procedures. Required soil parameters were given in the soil type and skin friction tables (t-z code 100).

- tzAPI1993 – t-z data will be generated by API-1993 procedures (t-z code 293).
- Q-Z curve types
  - qzManual – Manual input of q and z values for each layer and pile diameter (q-z code 0).
  - qzAutoLinear Automatic generation – linear (q-z code 100).
  - qzAutoBilinear Automatic generation – bilinear (q-z code 101).
  - qzAutoTrilinear Automatic generation – trilinear (q-z code 102).
  - Automatic generation as for p-y and t-z above. The q max value was given as SIG-TIP and the required displacement as DSPQZ/DIAM.
  - qzElasticTheory – A bilinear q-z curve is generated. Maximum tip stress is taken as SIG-TIP. Required displacement to develop SIG-TIP is calculated from elastic half space theory using the GO-SOIL and POIS values given (q-z code 200).
  - qzAPI1993 – q-z data will be generated by API-1993 procedures (q-z code 293)

## Soil Data

**Figure 5.23 The Create/Edit Soil Properties > Soil Data dialog**

- Initial value of soil shear modulus – Used to calculate the initial slope of the t-z curves and the q-z curves for the case of codes 200.
- Soil Poisson’s ratio – Used together with GO-SOIL to determine a modulus of elasticity for the calculation of initial slope of q-z curves for codes 200.
- Skin friction
- Peak skin friction in compression

- Peak skin friction in tension
- Ratio between displacement to reach peak skin friction and pile diameter – Used to find initial t-z slopes for codes 100.
- Tip resistance
- Peak tip stress acting at pile tip. This value may be governed either by soil strength or by the capacity of the internal soil plug in the case of an open ended pile.
- Ratio between displacement to reach peak tip stress and pile diameter – Used to find initial slope of q-z curves for codes 100.

### 5.3 The Hydro Properties

The hydro properties are listed in Table 5.1. To take effect in a wave load analysis a property must be assigned to a part of or the complete structure.

The properties Morison, Air Drag and Marine Growth may be functions of one or more of the variables Reynolds number, KC number, roughness, diameter and Z level. These methods of definition are categorised as A or B as shown by Table 5.1. Either one or both of these categories may be used for a certain property. However, you are not allowed to use two different methods of definition of the same category. For example, you may define the Morison property both as a constant and as a function of diameter but you cannot define it as function of both Rn and diameter.

For some of the properties only one property of that type can be defined and assigned to the structure, e.g. Morison coefficients as a function of the diameter. This means that it is not possible to define for instance two different functions describing the Morison coefficient as a function of the diameter and then assign them to different parts of the structure.

**Table 5.1 Hydro properties**

Type of property	Category	Mandatory / Optional	Number of properties
Morison, constant value	A	Mandatory for the wet part	Arbitrary
Morison, f(Rn,Rough)	B		1
Morison, f(KC,Rough)	B		1
Morison, f(Diameter)	B		1
Morison by rule	B		1
Air Drag, constant value	A	Mandatory for the dry part when calculating wind loads	Arbitrary
Air Drag, f(Rn)	B		1
Hydrodynamic Diameter		Optional	Arbitrary
Flooding		Optional	Arbitrary
Marine Growth, constant	A	Optional	Arbitrary
Marine Growth, f(Z)	B		Arbitrary

Table 5.1 Hydro properties

Type of property	Category	Mandatory / Optional	Number of properties
Element Refinement		Optional	Arbitrary
Conductor Shielding		Optional	Arbitrary

The optional properties Flooding and Element Refinement will be set by Wajac even if they are not explicitly assigned in GeniE. If no flooding property is assigned then Wajac will consider the structure to be non-flooded. If no refinement property is assigned Wajac will use a refinement factor of 1.0. How to apply a property to the structure is described in the GeniE User Manual Vol. 1.

**Note:** In Wajac it is possible to specify a Constant Morison Coefficients value between two Z levels by using the COEF command. In GeniE this will be handled by assigning the constant Morison coefficient property (Figure 5.25) to the part of the structure between these two Z levels. The COEF command will, therefore, never be generated by GeniE. This way of assigning a property is also valid for all 'constant value' hydro properties.

**Note:** Relevant parts of the structure must be selectable either as a predefined set or by rubberband selection. This in turn implies that the user must have the wave load analysis in mind in the modelling process, e.g. by splitting the structure at relevant Z-levels. Below is an example of how two marine growth properties are assigned to different levels. There is also an example showing a flooding property assigned to the flooded members.

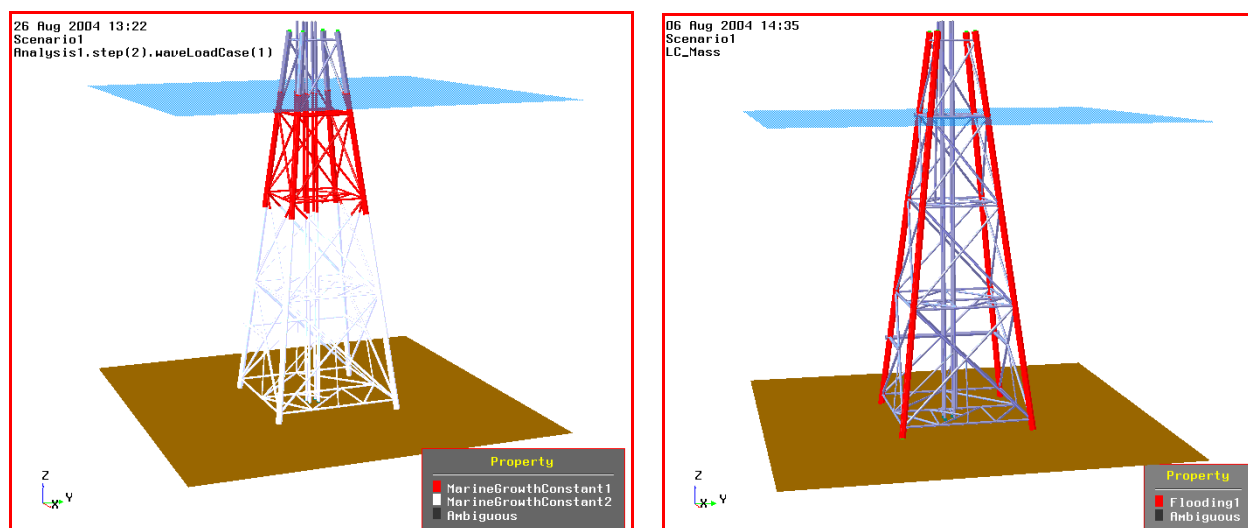


Figure 5.24 Examples of assignment of hydro properties

### Define Constant Morison Coefficients

The drag and inertia values are given in the element local system. Any number of this property may be defined and assigned to different parts of the structure. If some other Morison property than Constant is defined and assigned an overlapping part of the structure then the constant property will take precedence.

**Note:** Wajac will not use a Morison property that is assigned to the dry part of the structure. Morison Coefficients properties may, therefore, be assigned to the complete structure.

**Note:** The longitudinal coefficients  $C_{dx}$  and  $C_{mx}$  will only be used when calculating total structural loads as reported on the Wajac print file. No longitudinal force components will be written to the Loads Interface Files.

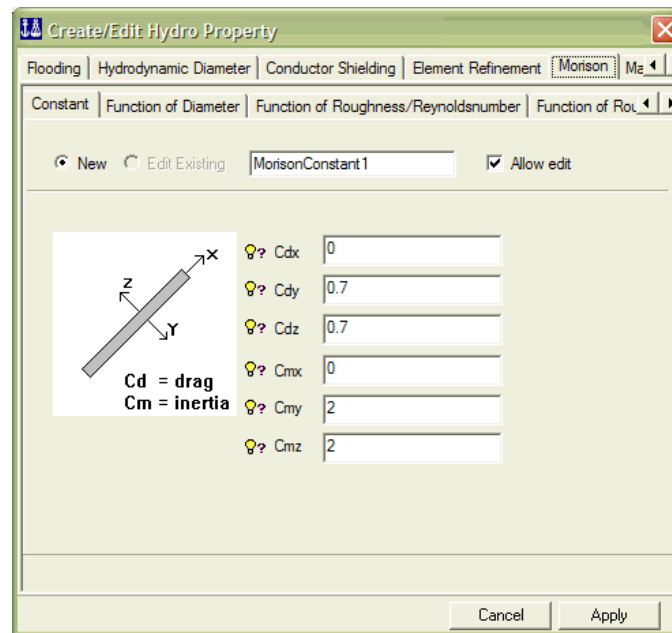


Figure 5.25 The Create/Edit Hydro Property > Morison > Constant dialog

### Define Morison coefficients by Rule

If this type of Morison coefficient property is assigned then the Rule tab of the Edit Wave Load Run dialog (Figure 5.45) must be set to API rules. Only one such property may be assigned.

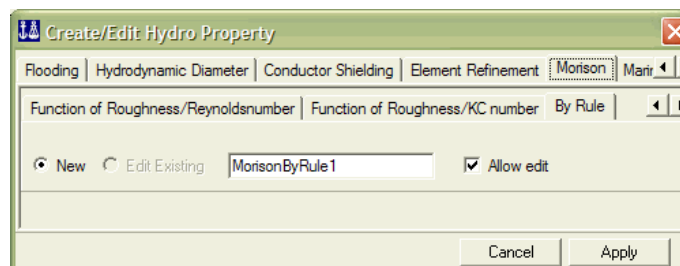


Figure 5.26 The Create/Edit Hydro Property > Morison > By Rule dialog



### Define Morison coefficients as function of diameter

Specify member diameter and corresponding normal component of hydrodynamic drag and inertia.

**Note:** The member diameter should include specified marine growth and member diameters should be specified in increasing order.

**Note:** The user should specify the entire range of diameters in the given structure. If a member diameter is outside the specified range the value closest to the diameter under consideration will be used. Wajac can handle a maximum of 12 diameter values. Only one such property may be assigned to the structure.

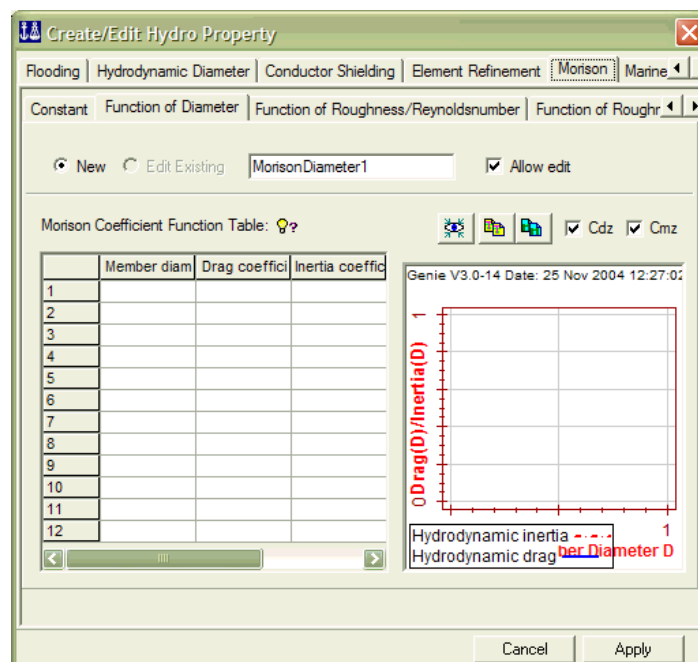
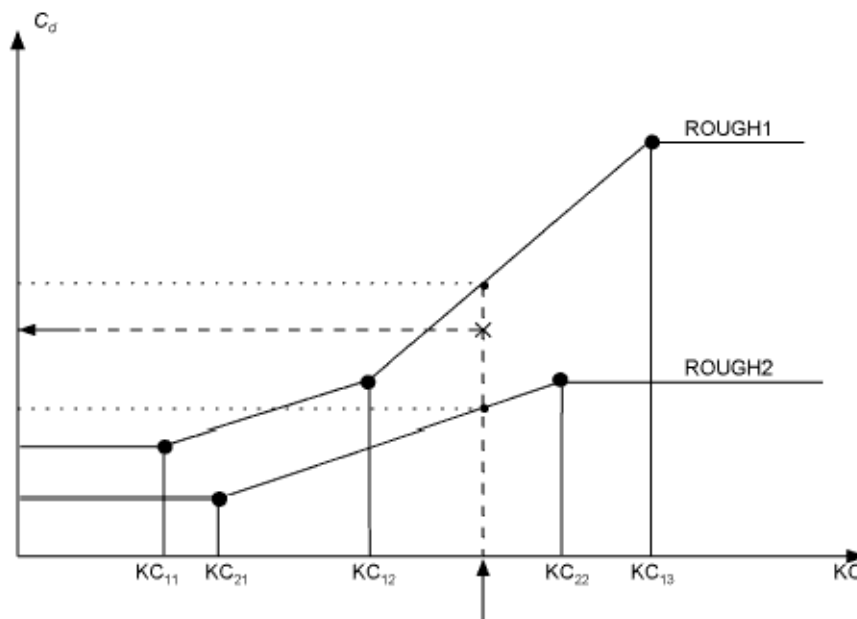


Figure 5.27 The Create/Edit Hydro Property > Morison > Function of Diameter dialog

### Define Morison coefficients as function of roughness and Reynolds number

The dialog has a Fill tool area. This will help the user define this property that is characterised by a number of roughness-value graphs consisting of KC/ Reynolds numbers as illustrated in Figure 5.28.



**Figure 5.28 Morison Coefficients as function of roughness and KC number**

This property is not applicable for seastate simulations or for calculation of transfer functions using equivalent linearisation. The command should not be used together with the API command.

Fill tool area:

- The number of rows for each roughness number must be given in the Repetitions column.
- When the Fill table button is clicked roughness numbers are automatically filled in the Morison Coefficient Function Table.
- Wajac can handle a maximum of 5 different roughness numbers.

Morison Coefficient Function Table:

- Specify Reynolds number and corresponding normal component of hydrodynamic drag and inertia.
- Reynolds numbers within one roughness number should be specified in increasing order.
- Wajac can handle a maximum of 9 Reynolds numbers for each roughness property.

Only one such property may be assigned to the structure.

The user should specify the entire range of Reynolds- and/or Keulegan-Carpenter numbers that will apply to the analysis. If the value at a load calculation point is outside the specified range the coefficient specified for the closest number will be used.

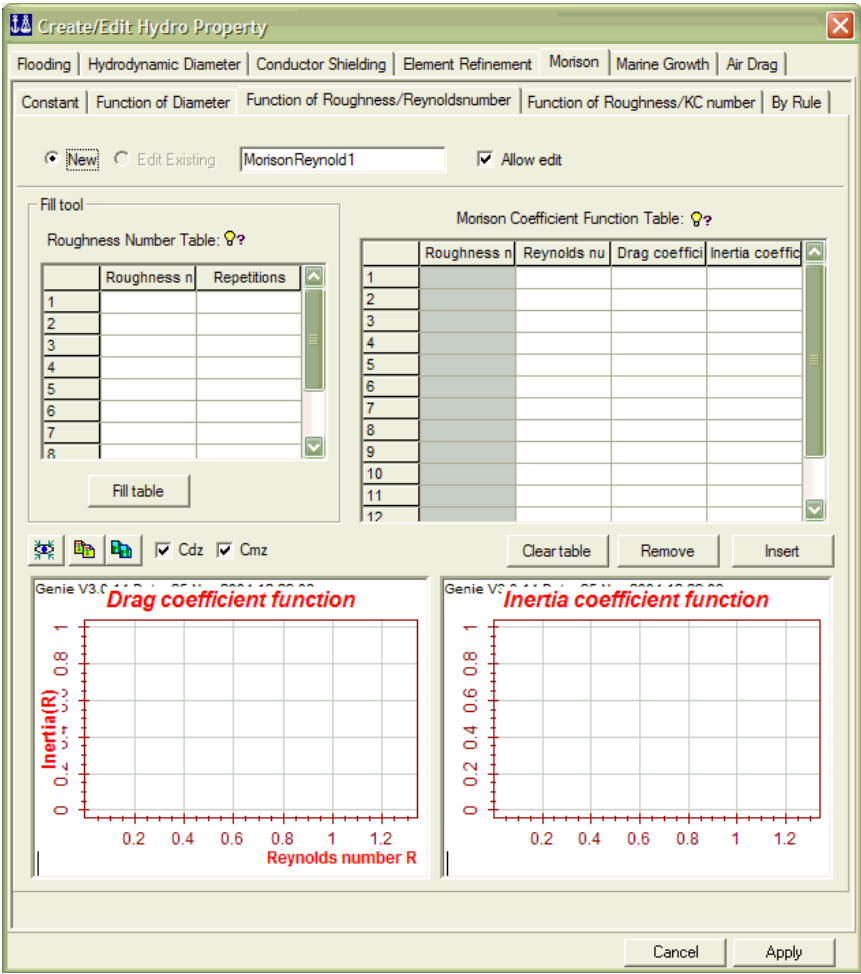


Figure 5.29 The Create/Edit Hydro Property > Morison > Function of Roughness/Reynoldsnumber dialog

Define Morison coefficients as function of roughness and Keulegan-Carpenter (KC) number

This property is defined in a way as explained above in Define Morison coefficients as function of roughness and Reynolds number.

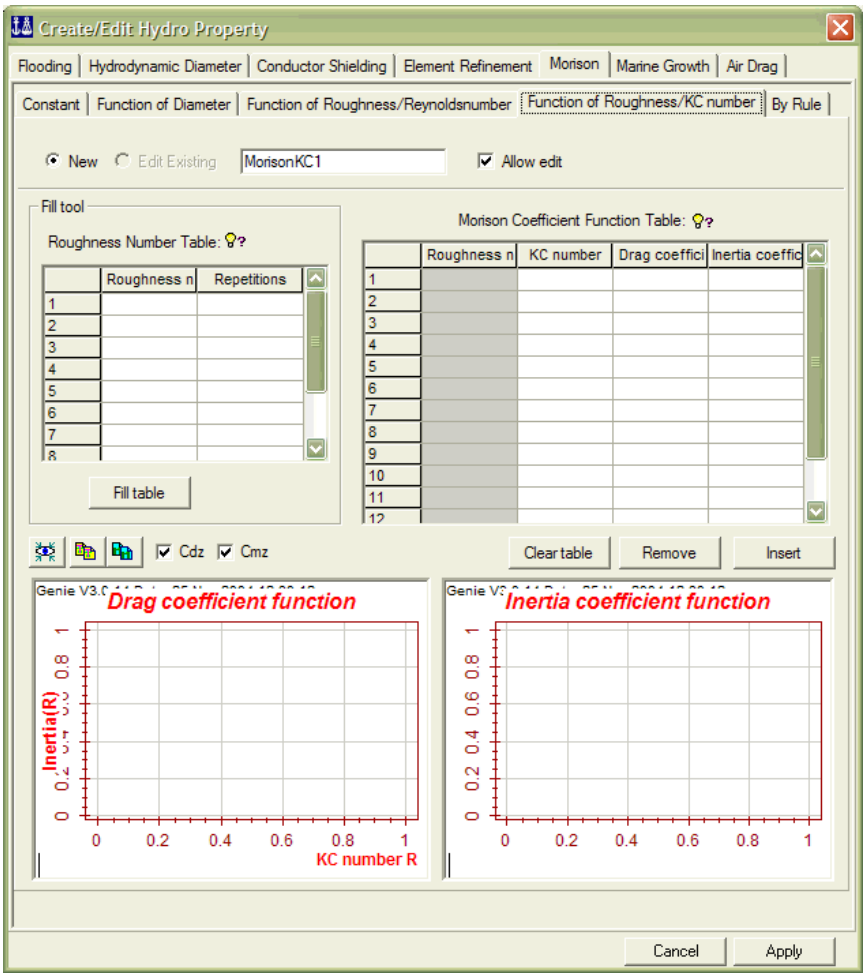


Figure 5.30 The Create/Edit Hydro Property > Morison > Function of Roughness/KC number dialog

## Define Flooding

Any number of this property may be assigned. It should be assigned to the wet and flooded part of the structure.

**Note:** If no flooding property is assigned Wajac assumes the structure to be non-flooded.

Wajac accepts only completely flooded (Flooding = 1) or non-flooded (Flooding = 0). Any other value will be converted to the nearest of the two. For a beam defined as flooded only the steel volume will contribute to the buoyancy.

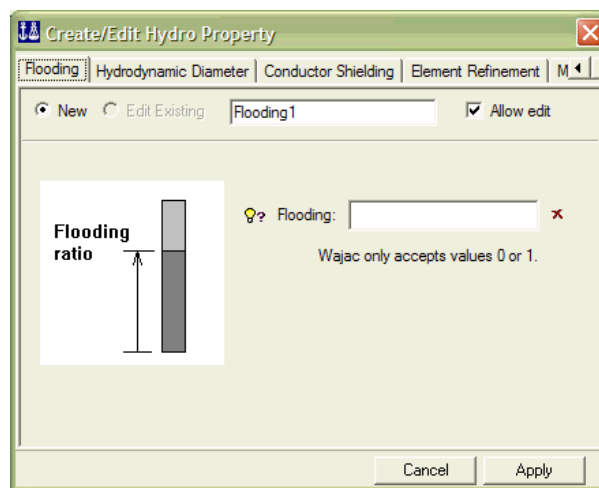


Figure 5.31 The Create/Edit Hydro Property > Flooding dialog

### Define Constant Air Drag Coefficients

The air drag values are given in the element local system. Any number of this property may be defined and assigned to different parts of the structure. If the Air Drag as function of Reynolds number property is defined and assigned to an overlapping part of the structure then the constant property will take precedence.

**Note:** Wajac will not use an Air Drag property that is assigned to the part of the structure below the SWL (still water level). Air Drag Coefficients properties may, therefore, be assigned to the complete structure.

**Note:** If the Prepare for gust wind induced fatigue option is selected in the Edit Wave Load Run dialog (Figure 5.43) then  $C_{dx}$  must be equal to  $C_{dz}$ .

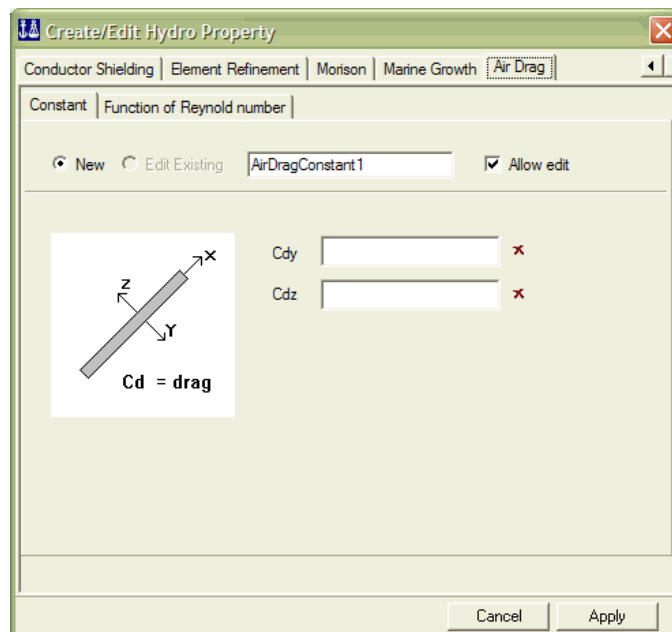


Figure 5.32 The Create/Edit Hydro Property > Air Drag > Constant dialog

Define Air Drag coefficients as function of Reynolds number

Specify air drag coefficients as a function of Reynolds number in the element local coordinate system. Reynolds numbers should be specified in increasing order.

The user should specify the entire range of Reynolds numbers that will apply to the analysis. If the value at a load calculation point is outside the specified range the coefficient specified for the closest number will be used.

**Note:** Wajac can handle a maximum of 21 Reynolds numbers.

**Note:** If the Prepare for gust wind induced fatigue option is selected in the Edit Wave Load Run dialog (Figure 5.43) then each Cdx must be equal to the corresponding Cdz.

Only one such property may be assigned. It may be assigned to the complete structure but preferably only to the dry part.

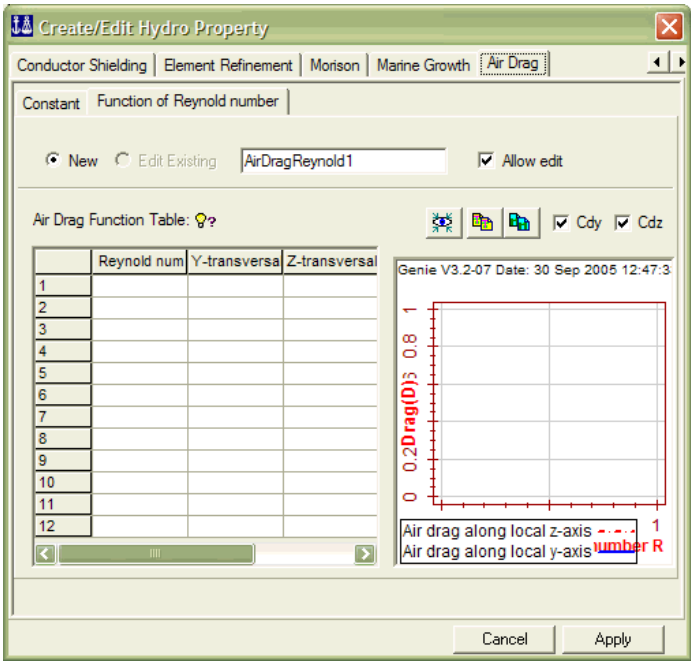


Figure 5.33 The Create/Edit Hydro Property > Air Drag > Function of Reynolds number dialog

### Define Constant Marine Growth

Any number of this property may be defined and assigned to the wet part of the structure. If the Marine Growth as function of Z-level property is defined and assigned to an overlapping part of the structure then the constant property will take precedence.

**Note:** Wajac will not use a Marine Growth property that is assigned to the dry part of the structure. Marine Growth properties may, therefore, be assigned to the complete structure.

Marine growth thickness is specified as an additional thickness. The marine growth times two will be added to the beam cross sectional diameter (equivalent diameter for non-pipe sections) before calculating the drag forces and, optionally, the inertia forces and added mass at load calculation points within the specified range. The roughness height is used to determine the roughness number (= roughness height / diameter) which in turn is used to specify varying hydrodynamic coefficients. The density factor contributes to the marine growth density (MGD) as  $MGD = \text{'density factor'} * RHO$ , where RHO is the water density. An increase in the marine growth density will reduce the buoyancy and increase the added mass (used in structural dynamics).

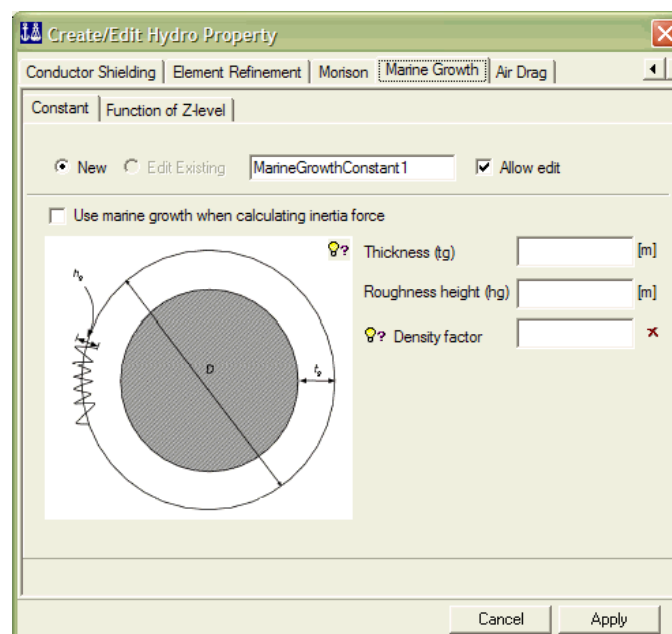


Figure 5.34 The Create/Edit Hydro Property > Marine Growth > Constant dialog



## Define Marine Growth coefficients as function of Z-level

Specify Z level and corresponding thickness, roughness height and density factor. The Z-coordinates are zero at the still water level and the Z-axis points upwards.

**Note:** The elevations must be given in sequence of decreasing Z-values.

The thickness, roughness height and density factor values are explained above in Define Constant Marine Growth. An arbitrary number of such properties may be assigned. It may be assigned to the complete structure but the Z-levels should be restricted to the wet part.

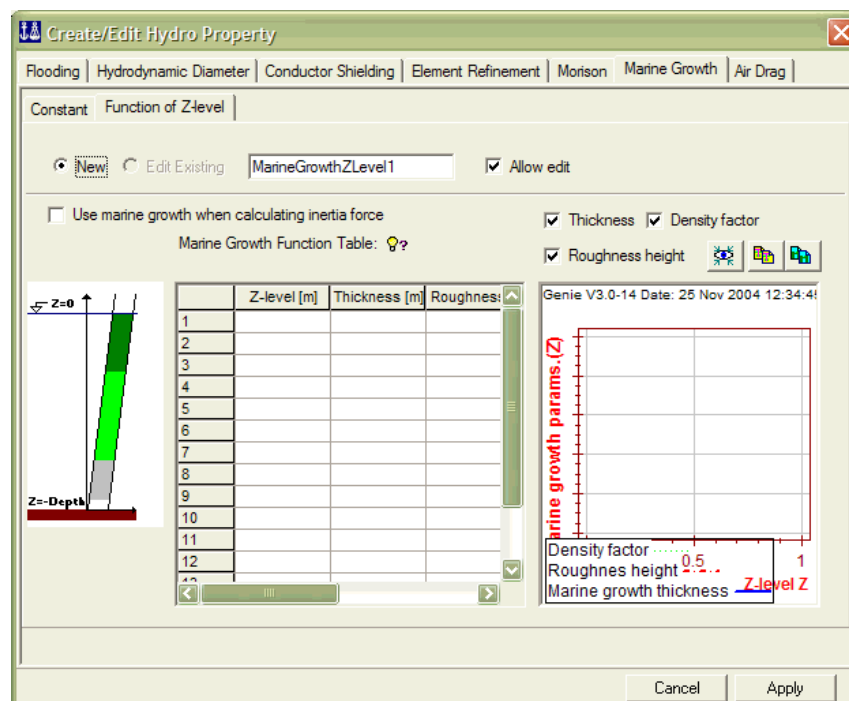


Figure 5.35 The Create/Edit Hydro Property > Marine Growth > Function of Z-level dialog

## Define Element Refinement

The following information is needed when defining this property.

Refinement coefficient:

- The default value is 1.0 and equals the default value of Wajac. The default value involves that only members longer than three times the wave length will be split into three or more segments.
- Irrespective of the refinement coefficient value the number of subelements will be minimum 2 and maximum 20.

- Value 0.0 is illegal. However, by letting the refinement coefficient be close to zero, the number of subelements will most likely be equal to the minimum value.

Maximum/Minimum subelement length:

- The default value is 0.0. Wajac will ignore this value.
- The maximum subelement length is 1/2 of the element length.
- The minimum subelement length is 1/20 of the element length.

The number of segments in each member is controlled by

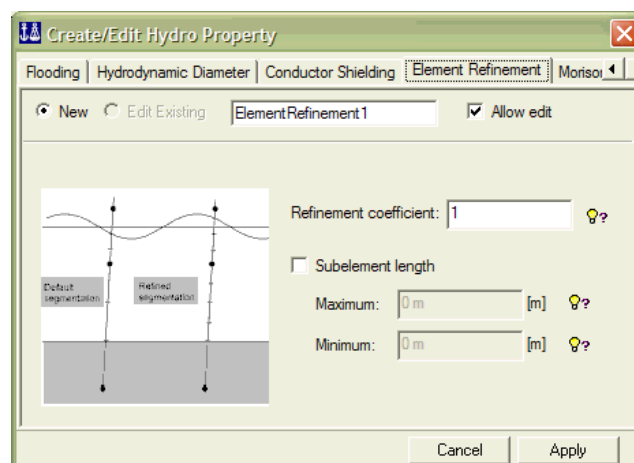
$$\text{Number of segments} = \frac{\text{SEG} \cdot \text{Member length}}{\text{Wave length}}$$

where SEG is the refinement coefficient.

Normally, high refinement is preferred close to the water surface where the variation in wave loads is high and low refinement is preferred close to seabed. The refinement property may be skipped since Wajac handles the refinement by default. Only in special cases demanding a high accuracy of the wave loads an element refinement higher than default need to be defined.

**Note:** An increased element refinement increases the size of the Loads Interface File and the Results Interface File.

An arbitrary number of such properties may be assigned. The Element Refinement property should be assigned in the same way as Constant Morison coefficients, i.e. between two Z-levels.

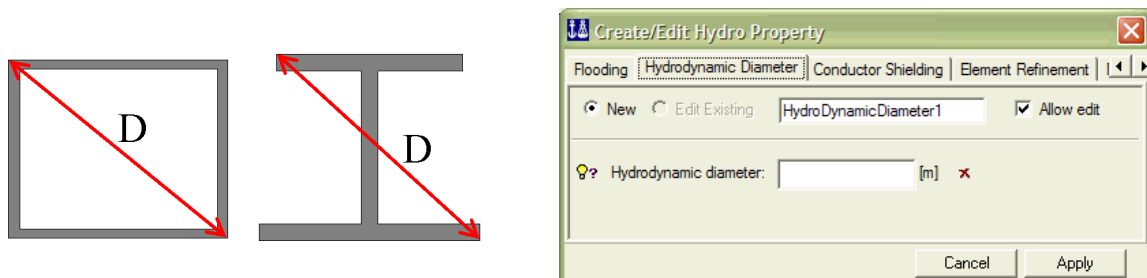


**Figure 5.36** The Create/Edit Hydro Property > Element Refinement dialog

## Define Hydrodynamic Diameter

An arbitrary number of such properties may be assigned. These properties may be assigned to the complete structure.

- The hydrodynamic diameter will be used for the structural parts it is assigned to as a replacement for the default value.
- For a pipe cross section the default value is the outer diameter.
- For non-circular cross sections the default value is the largest cross sectional dimension termed the 'equivalent diameter'.
- Diameters defined by this command have no effect on the buoyancy calculations.

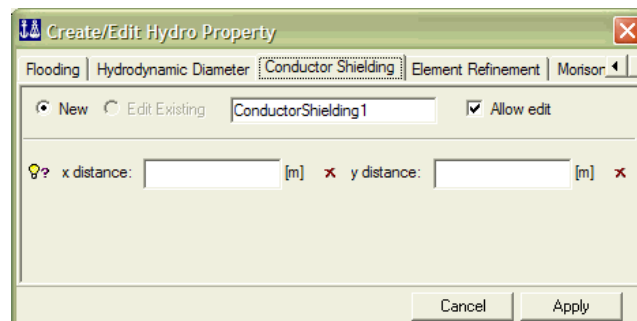


**Figure 5.37 The Create/Edit Hydro Property > Hydrodynamic Diameter dialog**

## Define Conductor Shielding

An arbitrary number of such properties may be assigned. These properties may be assigned to the complete structure.

- The averaged centre to centre distance between conductors is given in global X- and Y-directions.
- The distances are used to calculate shielding factors (by Wajac).
- It will take effect on specified and calculated drag and inertia coefficients for all elements for which this property has been assigned.



**Figure 5.38** The Create/Edit Hydro Property > Conductor Shielding dialog

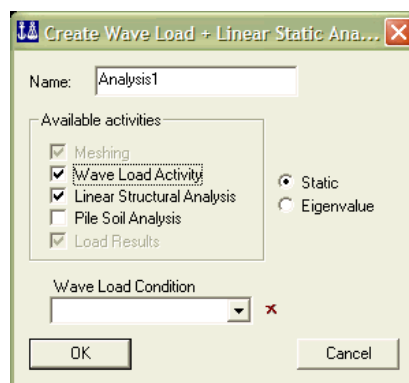
## 5.4 The Wave Load Analysis

A wave load analysis may either be a stand-alone activity or combined with a linear structural analysis or a structure-pile-soil analysis. An activity consists of several steps of which one is the Wave Load Analysis.

### 5.4.1 The Analysis Activity

The activity defines the relevant steps of which one is the Wave Load Analysis.

To perform a Design Wave Analysis an activity must first be created. There are three wave load related activities available: Wave Load, Wave Load + Linear Sestra Analysis and Wave Load + Pile Soil Analysis. To create an activity right-click the Analysis > Activities folder and select New Analysis. In the dialog that appears the Linear Structural Analysis option is already selected. Tick the wanted analysis and select an existing wave load condition in the Wave Load Condition combobox.



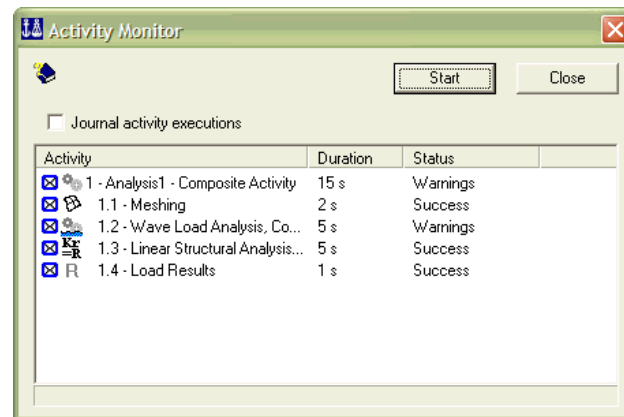
**Figure 5.39 The Create Analysis dialog**

When the Edit Wave Load Run is complete the Activity Monitor is used to determine analysis steps to be executed. Right-click the Analysis > Activities > Analysis1 object and select Activity Monitor. The Activity Monitor dialog (Figure 5.40) appears.

Features of the Activity Monitor dialog are:

- All activities selected in the list will be executed when the Start button is clicked.
- Right-click an activity to edit the activity.
- Right-click an activity to access result files and other relevant actions.
- The Start button starts the execution.

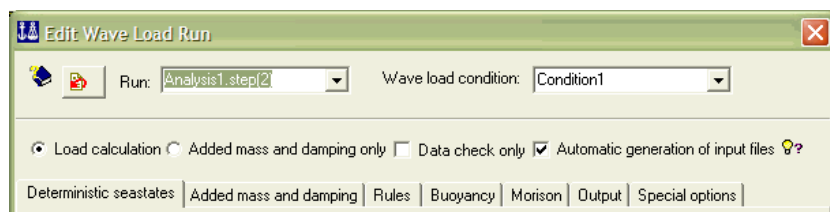
Before starting the execution complete the definition of the Wave Load Analysis activity. Open the Edit Wave Load Run dialog by right-clicking activity 1.2 and selecting Edit activity.



**Figure 5.40 The Activity Monitor dialog**

### 5.4.2 The Wave Load Run

The Edit Wave Load Run dialog consists of a common part plus seven tabs as shown in Figure 5.41. The Output tab contains three tabs. The data to be specified in these tabs will to some degree depend on the choice of analysis type and environment data. A description of the dialog in general and of each tab is provided below.



**Figure 5.41 The common part of the Edit Wave Load Run dialog and tabs**

### Analysis Type

The analysis type must be selected:

- The load calculation can currently only be of type deterministic time. Frequency domain calculation will be added in a later version of GeniE. The type of load calculation is determined by the wave load condition dataset selected.
- The Added mass and damping only option implies that no loadcases will be calculated and written to the Loads Interface File by Wajac. Only added mass and damping will be written.
- If Data check only is ticked Wajac will do a data check only. The user should check the Wajac.lis file as shown by Figure 3.17.
- The Wajac.inp file is created by default. The Automatic generation of input files option should be unchecked if the user wants to have the final control over the Wajac.inp file. In this case a Generate input

files button appears in the dialog; see Figure 5.42. Clicking this button will create an input file that may be manually edited. In other cases an old Wajac.inp file exists that the user may want to use.

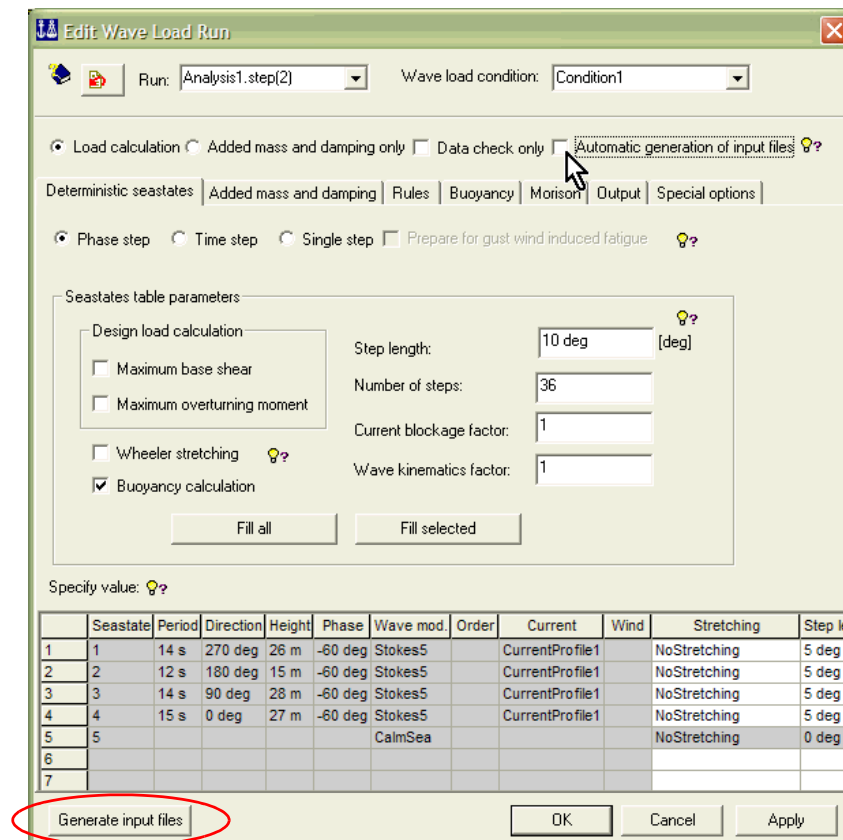


Figure 5.42 The common part of the Edit Wave Load Run dialog, plus all tabs

### The Deterministic seastates tab

This tab is only relevant when doing a Deterministic Time analysis. The seastate table is automatically filled with wave, current and wind data from the wave load condition selected. These values are not modifiable. The user must specify whether time, phase or single step should be used. This will apply to all seastates of a run. The Phase step option is default. If Time step option is selected then the Phase column of the seastate table changes to a Time column. If Single step is ticked the following happens:

- Irrelevant options in the Seastates table parameters area are disabled.
- Three columns of the seastate table are disabled: Step length, Num. steps and Design load.
- Default values are filled into disabled cells.
- The Prepare for gust wind induced fatigue checkbox is enabled.

**Note:** If a seastate has a CalmSea wave model the following columns are disabled: Stretching, Step length, Num. steps, Design load and Wave k. fac. They are all given default values.

The tab has a Seastates table parameters area to help the user fill the seastate table:

- When Single step is selected, the Step length and Number of steps fields are disabled. So are also the Design load calculation options.
- Wheeler stretching:
  - For linear wave theory (Airy waves) both waves and current are stretched.
  - For non-linear wave theory only current is stretched.
- The Current blockage factor and Wave kinematics factor fields have 1 as default value, i.e. no blockage of current or wave kinematics.
- For an explanation of how the Fill all and Fill selected buttons functions see the Regular Wave Set dialog (Figure 5.17).

Values may be entered into the seastates table either via comboboxes or by entering values directly. The following values/options may be given/selected in each column:

- Select Wheeler stretching or no-stretching. Wheeler stretching may be specified for linear wave theory (Airy) or current in combination with all wave theories.
- Give step length and number of phase/time steps.
- Select buoyancy: on/off.
- Select Design load calculation: Maximum base shear or Maximum overturning moment. You may also select both or none (no design load will be calculated).
- Give current blockage factor.
- Give wave kinematics factor. The global wave kinematics factor will not be included.



Deterministic seastates | Added mass and damping | Rules | Buoyancy | Morison | Output | Special options

☒ Phase step ☐ Time step ☐ Single step ☐ Prepare for gust wind induced fatigue

Seastates table parameters

Design load calculation

☐ Maximum base shear

☐ Maximum overturning moment

☐ Wheeler stretching

☒ Buoyancy calculation

Step length: 0 deg [deg]

Number of steps: 1

Current blockage factor: 1

Wave kinematics factor: 1

Fill all Fill selected

Specify value:

	Seastate	Period	Direction	Height	Phase	Wave mod.	Order	Current	Wind	Stretch
1	1	12.3 s	0 deg	23.3 m	340 deg	Stokes5		CurrentProfile1	WindProfileRelDir1	
2	2	12.3 s	45 deg	23.3 m	340 deg	Stokes5		CurrentProfile1	WindProfileRelDir2	
3	3					CalmSea				
4										
5										
6										

Figure 5.43 Deterministic seastates tab

### The Added mass and damping tab

Added mass and damping are calculated and written to the Loads Interface File independently of the choice of load calculation type. The default value is not to calculate added mass and damping.

- Include internal water option involves that added mass is written to the Loads Interface File including internal water in flooded members.
- The Use  $C_m=2.0$  option involves that  $C_m = 2.0$  (added mass coefficient = 1.0) is used for all elements irrespective of the  $C_m$  values used in the force calculation.

Deterministic seastates | Added mass and damping | Rules | Buoyancy | Morison | Output | Special options

☐ Calculate added mass

☐ Include internal water

☐ Use  $C_m=2.0$

☐ Calculate damping

Figure 5.44 Added mass and damping tab

## The Rules tab

None is the default option. If API is selected the following options are available:

- Include Doppler effect involves that the Doppler effect may be accounted for by using apparent wave (API 2.3.1b1).
- Include effect of wake encounter involves that the effect of wake encounter may be included. There are two options:
  - The dependency of wake encounter will be applied to all members within 15 degrees of the vertical (API 2.3.1b7) as specified by API.
  - The dependency of wake encounter will be applied to all members within 15 degrees of the orbital plane of the wave kinematics.
  - If a maximum angle is given this value will substitute 15 degrees.
  - The effect of wake encounter may be adjusted for current (API C2.3.1b7).

Deterministic seastates | Added mass and damping | **Rules** | Buoyancy | Morison | Output | Special options |

☒ None

☒ API

☐ Include Doppler effect

☐ Include effect of wake encounter

☒ Vertical

☒ Orbital plane

Maximum angle:  [deg]

☐ Adjust for current

Figure 5.45 Rules tab

## The Buoyancy tab

Assuming horizontal free surface:

- Buoyancy is calculated assuming a horizontal surface above each member.
- The largest of the surface elevations above the member ends is used at both ends.
- This option is default and is only available in time domain.

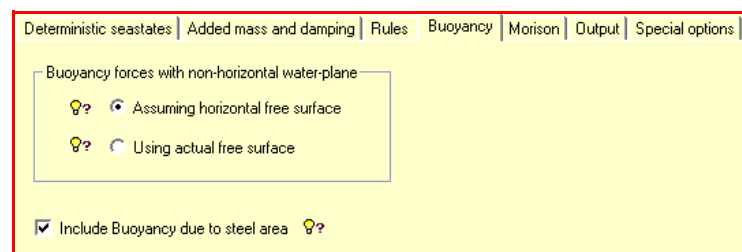


Figure 5.46 Buoyancy tab

Using actual free surface:

- Buoyancy forces are calculated using the actual, i.e. a non-horizontal waterplane that gives small buoyancy components in the horizontal plane.

By default the Include Buoyancy due to steel area option is on meaning that buoyancy of the steel is included.

## The Morison tab

Define the velocity decomposition used when calculating the normal components of the drag forces, i.e. handling of the  $v^2$  term in the Morison equation.

- Normal to element involves that  $\text{abs}(v_n)$  is used.
- Total involves that  $\text{abs}(v)$  is used.

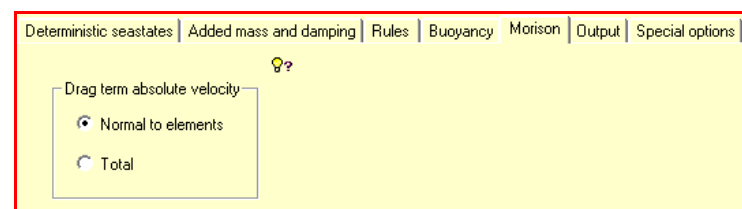


Figure 5.47 Morison tab

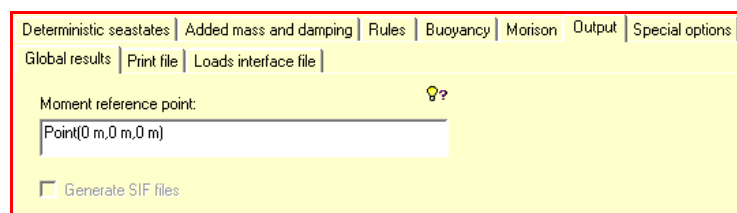
## The Output > Global results tab

Moment reference point:

- The moment reference point will influence global loads written to the Wajac.lis-file and the Simulation Result File; see Section 4.2.
- By default the origin of the global coordinate system is used.

Generate SIF files (currently not available as frequency domain analysis is not supported by the 3.2 version of GeniE):

- Generation of the Hydrodynamic Results Interface File, the G1.SIF file
- This option is only available when calculating transfer functions.
- By default no SIF file is generated.



**Figure 5.48 Output > Global results tab**

## The Output > Print file tab

This tab has two main areas: Element loads and Fluid kinematics. The Element loads area has two subareas: a Deterministic area that is enabled only in Deterministic Time mode and a Transfer functions area that is enabled only when calculating transfer functions.

For the Deterministic and Transfer functions comboboxes to be properly filled you need to click the Apply button.

In the Element loads area the Structure set combobox is used to select the part of the structure for which print of force intensities is desired. The selection will be included in the Add operation.

- Print of member loads is dependant on whether calculation of transfer functions or deterministic load is performed. In the comboboxes it will be possible to select All or a specified Seastate, Step/Direction and Period.
- Each time Add is clicked a new entry is appended to the table.

Fluid kinematics area:

- Print of fluid kinematics in specified points.
- Not available when calculating transfer functions.

- Each time Add is clicked a new entry is appended to the table.

Deterministic seastates
Added mass and damping
Rules
Buoyancy
Morison
Output
Special options

Global results
Print file
Loads interface file

Element loads

Structure set:

Deterministic

Seastate:
Step:

Add

	Structure set	Seastate	Step
1			
2			
3			
4			

Transfer functions

Direction:
Period:

Add

	Structure set	Direction	Period
1			
2			
3			
4			

Fluid kinematics

Point:

Add

	Points
1	
2	
3	
4	

Figure 5.49 Output > Print file tab

The Output > Loads interface file tab

The default option is to generate an unformatted Loads Interface File.

Specify first loadcase number:

- When checked the value decide the first loadcase number.
- If not checked the first loadcase number is automatically set to the highest existing loadcase number plus 1.

Deterministic seastates
Added mass and damping
Rules
Buoyancy
Morison
Output
Special options

Global results
Print file
Loads interface file

☒ Generate loads interface file

☐ Formatted files (ASCII)
☒ Unformatted files (binary)

☐ Specify first loadcase number:
0

Figure 5.50 Output > Loads interface file tab

## The Special options tab

The Use MacCamy-Fuchs' formula option is only available when calculating transfer functions.

- When In phase is selected the inertia forces from the solution are applied using a phase lag compared to the wave as given by the potential solution.
- When Phase lag is selected the inertia forces from the solution are applied in phase with the horizontal wave particle acceleration at the point in question.
- A On structure set may be selected and the solution will only be applied to this part of the structure. If no set is selected then solution is applied to the complete structure.

Eliminated structure:

- An Eliminated structure set may be selected. This part of the structure is eliminated from hydrodynamic load calculations. Default value is None.

Additional still water level:

- This option allows calculations to be performed for additional still water levels.
- The water levels must be relative to the still water level specified in the environment data.
- The table may for instance be used to include high and low tide.

The Exclude eccentricities in calculations option is by default off. If switched on the effect of member/element end eccentricities will be neglected.

Deterministic seastates | Added mass and damping | Rules | Buoyancy | Morison | Output | **Special options**

☒ Use MacCamy-Fuchs' formula ⓘ

☒ In phase

☒ Phase lag

☐ On structure:

☐ Eliminated structure:  ⓘ

☐ Exclude eccentricities in calculations

Additional still water level: ⓘ

	Still water level [m]
1	
2	
3	
4	
5	

**Figure 5.51 Special options tab**

## 5.5 Execution of a Wave Load Analysis

The aim of this section is to give some advice concerning execution of an analysis.

When a wave load analysis is executed the FEM-file (Input Interface File, see Section 4.2 about files used by GeniE) and Wajac.inp file are automatically generated. Global transfer functions calculated by Wajac may be written to the Results Interface File for statistical postprocessing in Postresp. Detailed loads are stored on the Loads Interface File for transfer to subsequent structural analysis in Sestra.

Error and warning messages are written to the Messages tab of GeniE and to the Wajac.lis and/or Wajac.inp files.

A mistake that may be done is to forget to complete the wave load run definition. The Activity Monitor will then display the text Error in the Status field and give an indication of what the error is through a text as shown in Figure 5.52. To solve this problem open the Edit Wave Load Run dialog and complete the definition.

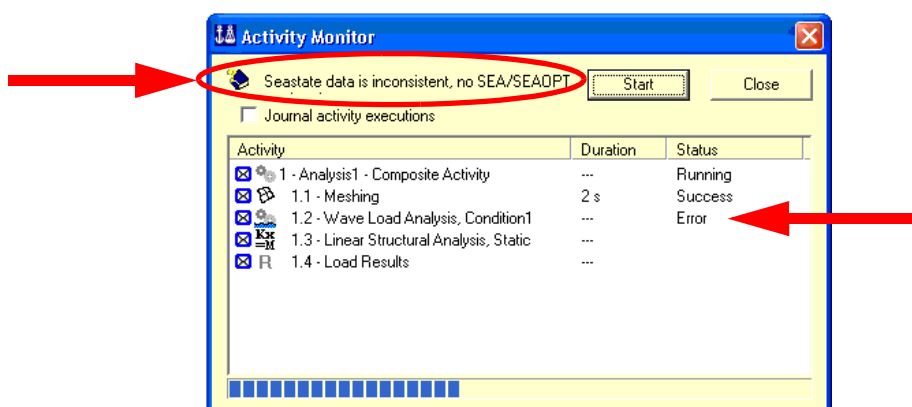


Figure 5.52 The Activity Monitor dialog with error message

**Note:** The Command Line view of GeniE refers to more information on errors in the Messages view as shown in Figure 5.53.

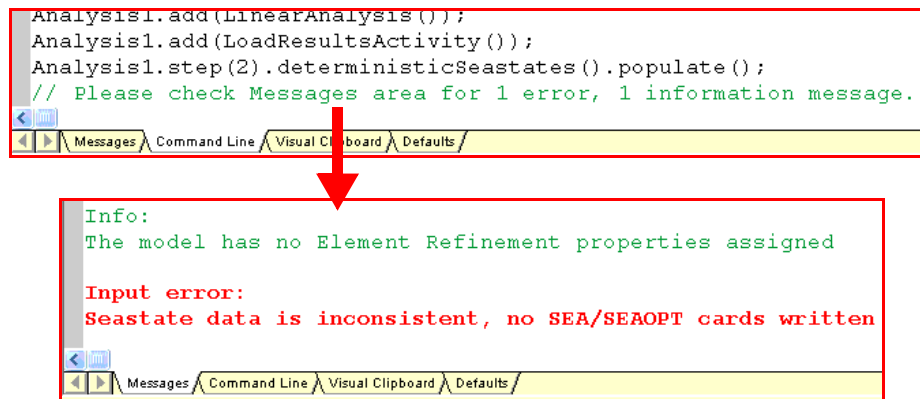


Figure 5.53 Information on errors in the Command Line and Messages view

Even when an execution has been completed there may be warnings. You should check these warnings. Right-click the Wave Load Analysis activity and select `wajac.lis`. The file appears in the text editor associated with the file type.

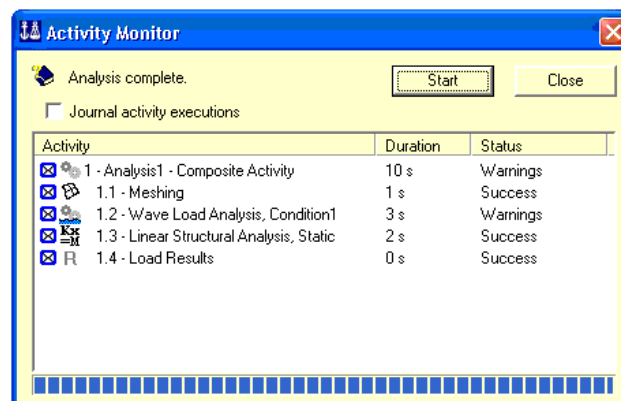


Figure 5.54 The Activity Monitor dialog after a successful analysis



## 5.6 Execution of a Structure-Pile-Soil Analysis

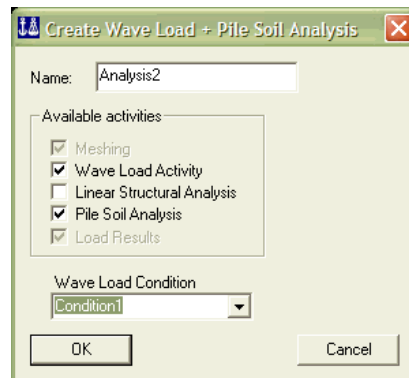


Figure 5.55 The Create Wave Load + Pile Soil Analysis dialog

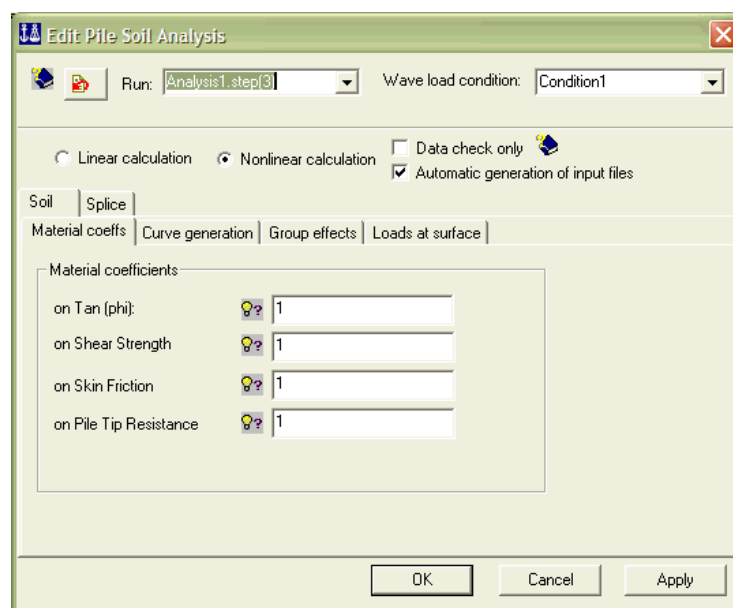
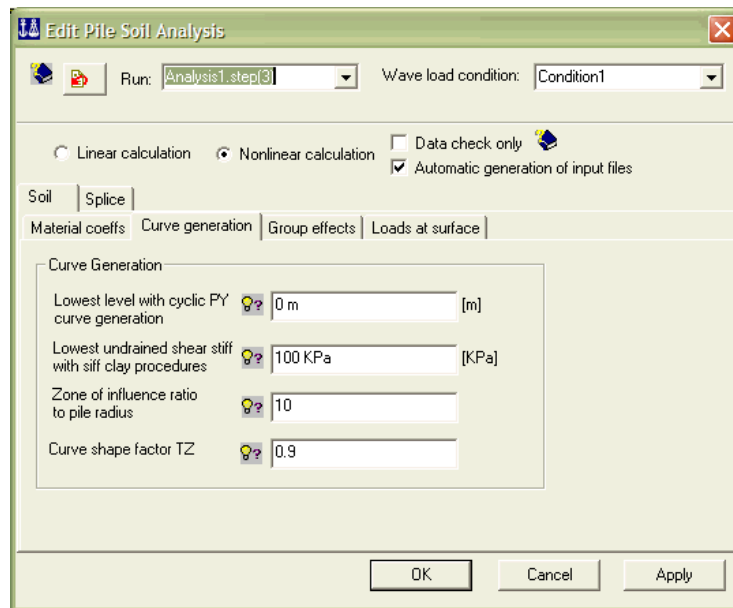


Figure 5.56 The Edit Pile Soil Analysis > Soil > Material coeffs dialog

These coefficients can be used to modify the soil properties given below without changing the actual values given.

The values given are divided by the coefficients, i.e. a coefficient greater than one reduces the capacity.

The material coefficient for the friction angle applies to  $\tan(\phi)$ .



**Figure 5.57 The Edit Pile Soil Analysis > Soil > Curve generation dialog**

- Lowest level with cyclic PY curve generation

Cyclic p-y data will be generated for all layers below the specified limit.

- Lowest undrained shear stiff(ness) with stiff clay procedures

The API code distinguishes between ‘soft clay’ and ‘stiff clay’. Procedures for calculating p-y data are, however, only given for soft clay. Gensod takes a clay layer to be ‘stiff’ when the undrained strength value given is higher than the limit specified. In that case the Reese et al. (1975) recommendations are used rather than the API ‘soft’ clay procedures.

The two values below are needed for the Kraft et al. (1981) procedures to generate t-z data, t-z code = 200.

- Zone of influence ratio to pile radius

Outside the zone of influence the shear displacements due to pile axial loading are neglected. Typical values may be 10 to 30.

- Curve shape factor TZ

Dimensionless factor that governs the shape of the t-z curve up to the peak resistance. Zero gives a linear curve. Typical values may be 0.8 to 0.99.

The screenshot shows the 'Edit Pile Soil Analysis' dialog box with the 'Soil' tab selected. Within this tab, the 'Group effects' sub-tab is active. The 'Run' dropdown is set to 'Analysis1\_step[3]' and the 'Wave load condition' is 'Condition1'. Under the 'Nonlinear calculation' radio button, there are checkboxes for 'Data check only' (unchecked) and 'Automatic generation of input files' (checked). The 'Modulus Of Elasticity' section contains two rows of input fields: the first row has '10000 KPa' and '0 m', and the second row has '10000 KPa' and '-100 m'. The 'Average Poisson Ratio' is set to '0.5'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

**Figure 5.58 The Edit Pile Soil Analysis > Soil > Group effects dialog**

Group interaction effects (pile-soil-pile interaction) are calculated from elastic half-space theory assuming an E modulus that increases linearly with depth.

The screenshot shows the 'Edit Pile Soil Analysis' dialog box with the 'Soil' tab selected. Within this tab, the 'Loads at surface' sub-tab is active. The 'Run' dropdown is set to 'Analysis1\_step[3]' and the 'Wave load condition' is 'Condition1'. Under the 'Nonlinear calculation' radio button, there are checkboxes for 'Data check only' (unchecked) and 'Automatic generation of input files' (checked). The 'Verical stress at soil surface' is set to '0 KPa'. The 'Embankment loads' section contains three rows of input fields: 'Verical stress under embakment' (0 KPa), 'Width of embankment sloping part' (0 m), and 'Distance pile to embankment toe' (0 m). The 'Circular loaded area' section contains two rows of input fields: 'Verical stress under loaded area' (0 KPa) and 'Radius of circular loaded area' (0 m). At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

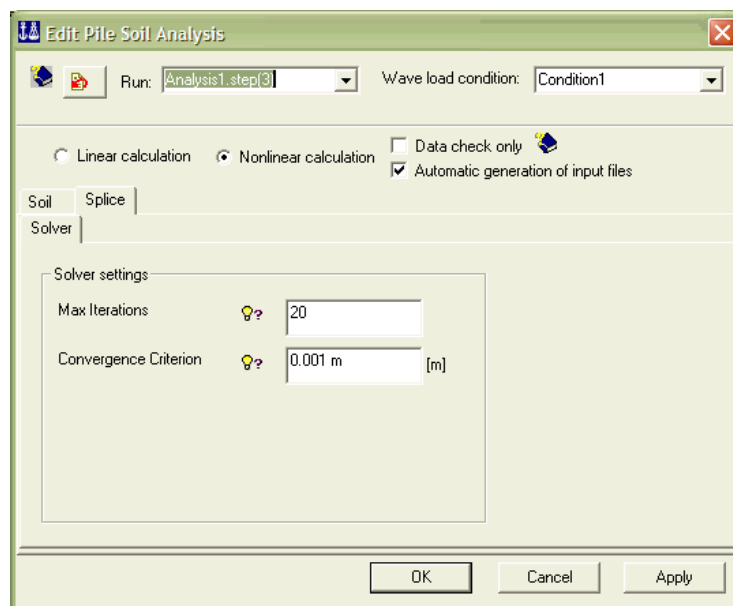
**Figure 5.59 The Edit Pile Soil Analysis > Soil > Loads at surface dialog**

Different types of vertical loading may exist at the soil surface in the vicinity of the pile. Such loading will influence the vertical stress down along the pile and hence the lateral resistance. Gensod approximately account for:

- A uniform Vertical stress at soil surface
- An embankment type loading
- A circular loaded area with the pile at its centre

The positions of the different types of surface loads are therefore given relative to a hypothetical average pile. Vertical loads are assumed to act at a Z level corresponding to the general scoured soil surface. The required inputs for each of the 3 options are:

- Vertical stress at soil surface:
  - Uniform vertical stress acting over a large area
- Embankment loads are defined by:
  - Vertical stress under the flat part of the embankment loading
  - Width of the sloping part of the embankment loading
  - Horizontal distance from the pile axis to the toe of the embankment. Positive if the pile is located outside the embankment, negative if the pile is inside.
- Circular loaded area is defined by:
  - Vertical stress under the circular loaded area
  - Radius of the circular loaded area



**Figure 5.60 The Edit Pile Soil Analysis > Splice > Solver dialog**

- **Max Iterations**

The iterations are stopped when the convergence criterion is met or when the maximum number of iterations is reached. Max value is 50.

- **Convergence Criterion**

The convergence criterion is met when the last iteration change in displacements along the pile is less than the specified length value.



## APPENDIX A TUTORIAL EXAMPLES

### A 1 Fixed 4-Legged Jacket Input for Design Wave Analysis

The input below creates the 4-legged jacket shown in Figure A.1. The model is used in the tutorial example of Section 3.2. Note that the part of the input defining wave load data has been commented out (all lines between the line starting with `/*` and the last line `*/`). Remove these two lines if you want to run through the complete example rather than working through the tutorial.

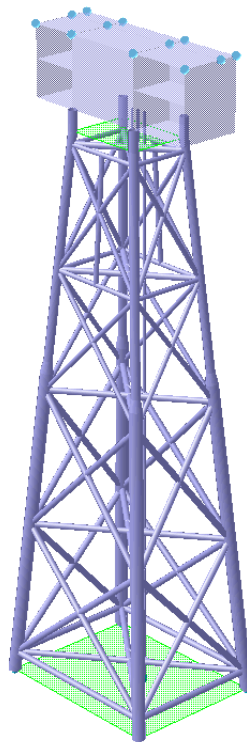


Figure A.1 4-legged jacket

```

// Fixed 4 Legged jacket with risers and simplified topside
// Complete input except that definition of wave load data is commented out, search for '/*' to find where this data starts
// Created by Aanund B. Berdal, DNV Software
// Revised Sept. 15, 2005
GenieRules.Units.setDatabaseUnits("m", "kN", "delC");
GenieRules.Units.setInputUnit(Length, "m");
GenieRules.Units.setInputUnit(Force, "kN");
GenieRules.Units.setInputUnit(TempDiff, "delC");
GenieRules.Tolerances.useTolerantModelling = true;
// Sections and material
Pipe06 = PipeSection(0.6,0.01);
Pipe12 = PipeSection(1.2,.03);
Pipe16 = PipeSection(1.6,.03);
Pipe21 = PipeSection(2.1,.08);
Pipe22 = PipeSection(2.2,.08);
Pipe32 = PipeSection(3.2,.09);
Steel = Material(3.56E5,7.85,2.1E8,0.3,0.12E-04,0);
Steel.setDefault();
// Create legs
Pipe32.setDefault();
Leg1 = Beam(Point(-22.5 m,-17.5 m,0 m),Point(-10 m,-7.5 m,138 m));
Leg2 = Beam(Point(22.5 m,-17.5 m,0 m),Point(10 m,-7.5 m,138 m));
Leg3 = Beam(Point(22.5 m,17.5 m,0 m),Point(10 m,7.5 m,138 m));
Leg4 = Beam(Point(-22.5 m,17.5 m,0 m),Point(-10 m,7.5 m,138 m));
// Fix lower end of legs
Sp4 = SupportPoint(Point(-22.5 m,-17.5 m,0 m));
Sp5 = SupportPoint(Point(22.5 m,-17.5 m,0 m));
Sp6 = SupportPoint(Point(22.5 m,17.5 m,0 m));
Sp7 = SupportPoint(Point(-22.5 m,17.5 m,0 m));
// Create bracing between elevations 5 and 75
Pipe16.setDefault();
Br01 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,-17.13768116 m,5 m));
Br02 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br03 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
Br04 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,-17.13768116 m,5 m));
Br05 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br06 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
Br07 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));
Br08 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br09 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br10 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));

```



```

Br11 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br12 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br13 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br14 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br15 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br16 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br17 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br18 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br19 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br20 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br21 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br22 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br23 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(18.87681159 m,14.60144928 m,40 m));
Br24 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br25 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(15.70652174 m,12.06521739 m,75 m));
Br26 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-15.70652174 m,12.06521739 m,75 m));
// Create bracing between elevations 75 and 135
Pipe12.setDefault();
Br27 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br28 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br29 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br30 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br31 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br32 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br33 = Beam(Point(10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br34 = Beam(Point(-10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br35 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br36 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br37 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));
Br38 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br39 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br40 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));
Br41 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br42 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br43 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br44 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br45 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br46 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br47 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br48 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br49 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br50 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));

```

```

Br51 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-9.891304348 m,105 m));
Br52 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-7.717391304 m,135 m));
Br53 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,9.891304348 m,105 m));
Br54 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,7.717391304 m,135 m));
// Create horizontal bracing at elevations 105 and 135
Pipe16.setDefault();
Br55 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br56 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br57 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br58 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
// Change section for upper legs
Cone1 = ConeSection(1.0,true);
Leg1.divideSegmentAt(1,0.575848871);
Leg1.divideSegmentAt(2,0.05091188302);
Leg1.SetSegmentSection(3,Pipe22);
Leg1.SetSegmentSection(2,Cone1);
Leg2.divideSegmentAt(1,0.575848871);
Leg2.divideSegmentAt(2,0.05091188302);
Leg2.SetSegmentSection(3,Pipe22);
Leg2.SetSegmentSection(2,Cone1);
Leg3.divideSegmentAt(1,0.575848871);
Leg3.divideSegmentAt(2,0.05091188302);
Leg3.SetSegmentSection(3,Pipe22);
Leg3.SetSegmentSection(2,Cone1);
Leg4.divideSegmentAt(1,0.575848871);
Leg4.divideSegmentAt(2,0.05091188302);
Leg4.SetSegmentSection(3,Pipe22);
Leg4.SetSegmentSection(2,Cone1);
// Create short stubs at top of legs
Pipe22.setDefault();
Col1 = Beam(Point(-10 m,-7.5 m,138 m),Point(-10 m,-7.5 m,138 m + 5 m));
Col2 = Col1.copyTranslate(Vector3d(20 m,0 m,0 m));
Col3 = Col1.copyTranslate(Vector3d(0 m,15 m,0 m));
Col4 = Col2.copyTranslate(Vector3d(0 m,15 m,0 m));
// Create simplified topside
Th2 = Thickness(0.02);
Th4 = Thickness(0.04);
Th6 = Thickness(0.06);
Th6.setDefault();
DeckLow = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,7.5 m,143 m),Point(-22 m,7.5 m,143 m));
DeckMid = DeckLow.copyTranslate(Vector3d(0 m,0 m,9 m));
DeckUpp = DeckLow.copyTranslate(Vector3d(0 m,0 m,17 m));

```

```

WallyZ1 = Plate(Point(-22 m,-7.5 m,143 m),Point(-22 m,7.5 m,143 m),Point(-22 m,7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallyZ2 = WallyZ1.copyTranslate(Vector3d(12 m,0 m,0 m));
WallyZ3 = WallyZ1.copyTranslate(Vector3d(32 m,0 m,0 m));
WallyZ4 = WallyZ1.copyTranslate(Vector3d(48 m,0 m,0 m));
WallXZ2 = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,-7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallXZ2.moveTranslate(Vector3d(0 m,10.5 m,0 m),geUNCONNECTED);
WallXZ1 = Plate(Point(-10 m,-7.5 m,143 m),Point(10 m,-7.5 m,143 m),Point(10 m,-7.5 m,160 m),Point(-10 m,-7.5 m,160 m));
WallXZ3 = WallXZ1.copyTranslate(Vector3d(0 m,15 m,0 m));
// Add mass points to upper deck of topside
Mass1 = PointMass(Point(-22 m,-7.5 m,160 m),600);
Mass2 = PointMass(Point(-22 m,3 m,160 m),600);
Mass3 = PointMass(Point(-22 m,7.5 m,160 m),600);
Mass4 = PointMass(Point(-10 m,-7.5 m,160 m),600);
Mass5 = PointMass(Point(-10 m,3 m,160 m),600);
Mass6 = PointMass(Point(-10 m,7.5 m,160 m),600);
Mass7 = PointMass(Point(10 m,-7.5 m,160 m),600);
Mass8 = PointMass(Point(10 m,3 m,160 m),600);
Mass9 = PointMass(Point(10 m,7.5 m,160 m),600);
Mass10 = PointMass(Point(26 m,-7.5 m,160 m),600);
Mass11 = PointMass(Point(26 m,3 m,160 m),600);
Mass12 = PointMass(Point(26 m,7.5 m,160 m),600);
// Create conductor supports and non-structural conductors
Pipe16.setDefault();
Cndsupp1 = Beam(Point(-3.992957746 m,3 m,135 m),Point(3.992957746 m,3 m,135 m));
Cndsupp2 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-30 m));
Cndsupp3 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-60 m));
Cndsupp4 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-95 m));
Cndsupp5 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-130 m));
Pipe06.setDefault();
Cnd1 = Beam(Point(-2 m,3 m,0 m),Point(-2 m,3 m,143 m));
Cnd2 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*1);
Cnd3 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*2);
Non = BeamTypeNonstruct();
Cnd1.beamType = Non;
Cnd2.beamType = Non;
Cnd3.beamType = Non;
// Fix lower ends of non-structural conductors that have no stiffness
Sp1 = SupportPoint(Point(-2 m,3 m,0 m));
Sp2 = SupportPoint(Point(0 m,3 m,0 m));
Sp3 = SupportPoint(Point(2 m,3 m,0 m));
// Create set named Jacket including all beams above mudline and up to lower deck of topside
Jacket = Set();

```

```
Jacket.add(Leg1);
Jacket.add(Leg2);
Jacket.add(Leg3);
Jacket.add(Leg4);
Jacket.add(Br01);
Jacket.add(Br02);
Jacket.add(Br03);
Jacket.add(Br04);
Jacket.add(Br05);
Jacket.add(Br06);
Jacket.add(Br07);
Jacket.add(Br08);
Jacket.add(Br09);
Jacket.add(Br10);
Jacket.add(Br11);
Jacket.add(Br12);
Jacket.add(Br13);
Jacket.add(Br14);
Jacket.add(Br15);
Jacket.add(Br16);
Jacket.add(Br17);
Jacket.add(Br18);
Jacket.add(Br19);
Jacket.add(Br20);
Jacket.add(Br21);
Jacket.add(Br22);
Jacket.add(Br23);
Jacket.add(Br24);
Jacket.add(Br25);
Jacket.add(Br26);
Jacket.add(Br27);
Jacket.add(Br28);
Jacket.add(Br29);
Jacket.add(Br30);
Jacket.add(Br31);
Jacket.add(Br32);
Jacket.add(Br33);
Jacket.add(Br34);
Jacket.add(Br35);
Jacket.add(Br36);
Jacket.add(Br37);
Jacket.add(Br38);
```

```
Jacket.add(Br39);
Jacket.add(Br40);
Jacket.add(Br41);
Jacket.add(Br42);
Jacket.add(Br43);
Jacket.add(Br44);
Jacket.add(Br45);
Jacket.add(Br46);
Jacket.add(Br47);
Jacket.add(Br48);
Jacket.add(Br49);
Jacket.add(Br50);
Jacket.add(Br51);
Jacket.add(Br52);
Jacket.add(Br53);
Jacket.add(Br54);
Jacket.add(Br55);
Jacket.add(Br56);
Jacket.add(Br57);
Jacket.add(Br58);
Jacket.add(Col1);
Jacket.add(Col2);
Jacket.add(Col3);
Jacket.add(Col4);
Jacket.add(Cndsupp1);
Jacket.add(Cndsupp2);
Jacket.add(Cndsupp3);
Jacket.add(Cndsupp4);
Jacket.add(Cndsupp5);
Jacket.add(Cnd2);
Jacket.add(Cnd1);
Jacket.add(Cnd3);
// Create set named Legs including the four legs
Legs = Set();
Legs.add(Leg2);
Legs.add(Leg1);
Legs.add(Leg3);
Legs.add(Leg4);
// Create set named Topside including decks and walls of topside
Topside = Set();
Topside.add(DeckUp);
Topside.add(DeckMid);
```

```

Topside.add(DeckLow);
Topside.add(WallyZ1);
Topside.add(WallyZ2);
Topside.add(WallyZ3);
Topside.add(WallyZ4);
Topside.add(WallXZ1);
Topside.add(WallXZ2);
Topside.add(WallXZ3);
// Define mesh density for topside, feature edge used to control mesh
Mesh1 = MeshDensity(20);
Topside.meshDensity = Mesh1;
FEdge1 = FeatureEdge(Point(-2 m,-7.5 m,143 m),Point(-2 m,7.5 m,143 m));
FEdge2 = FeatureEdge(Point(-2 m,-7.5 m,152 m),Point(-2 m,7.5 m,152 m));
FEdge3 = FeatureEdge(Point(-2 m,-7.5 m,160 m),Point(-2 m,7.5 m,160 m));
FEdge4 = FeatureEdge(Point(-2 m,3 m,143 m),Point(-2 m,3 m,160 m));
FEdge5 = FeatureEdge(Point(0 m,-7.5 m,143 m),Point(0 m,7.5 m,143 m));
FEdge6 = FeatureEdge(Point(0 m,-7.5 m,152 m),Point(0 m,7.5 m,152 m));
FEdge7 = FeatureEdge(Point(0 m,-7.5 m,160 m),Point(0 m,7.5 m,160 m));
FEdge8 = FeatureEdge(Point(0 m,3 m,143 m),Point(0 m,3 m,160 m));
FEdge9 = FeatureEdge(Point(2 m,-7.5 m,143 m),Point(2 m,7.5 m,143 m));
FEdge10 = FeatureEdge(Point(2 m,-7.5 m,152 m),Point(2 m,7.5 m,152 m));
FEdge11 = FeatureEdge(Point(2 m,-7.5 m,160 m),Point(2 m,7.5 m,160 m));
FEdge12 = FeatureEdge(Point(2 m,3 m,143 m),Point(2 m,3 m,160 m));
// Create gravity and wind load cases
Gravity = LoadCase();
Gravity.setAcceleration(Vector3d(0 m/s^2,0 m/s^2,-9.80665 m/s^2));
Gravity.includeSelfWeight();
WindN = LoadCase();
Wind1 = FlatSurfacePressure(WindN,5 KPa,Point(-22,3,143),Point(26,3,143),Point(26,3,160),Point(-22,3,160));
WindE = LoadCase();
Wind2 = FlatSurfacePressure(WindE,7 KPa,Point(26,-7.5,143),Point(26,-7.5,160),Point(26,7.5,160),Point(26,7.5,143));
WindS = LoadCase();
Wind3 = FlatSurfacePressure(WindS,6 KPa,Point(-22,3,143),Point(-22,3,160),Point(26,3,160),Point(26,3,143));
WindW = LoadCase();
Wind4 = FlatSurfacePressure(WindW,9 KPa,Point(-22,7.5,143),Point(-22,7.5,160),Point(-22,-7.5,160),Point(-22,-7.5,143));
/* The rest of the input is commented out
// Here hydro properties starts
// Define location
Location1 = Location();
Location1.gravity = 9.80665 m/s^2;
Location1.air.density = 0.001226 tonne/m^3;
Location1.air.kinematicViscosity = 1.492e-008 kN*s/m^2;

```

```

Location1.water.density = 1.025 tonne/m^3;
Location1.water.kinematicViscosity = 1.981e-009 kN*s/m^2;
Location1.waterSurface.z = 124 m;
Location1.seabed.normaldirection = Vector3d(0 m,0 m,1 m);
Location1.seabed.z = 0 m;
// Define current profile
CP1_Elevations = Array(30,0,-44,-124);
CP1_Directions = Array(0 deg,0 deg,0 deg,0 deg);
CP1_Velocities = Array(1.1,1.1,0.5,0);
CurrentProfile1 = CurrentProfileRelDir(CP1_Elevations,CP1_Directions,CP1_Velocities,dtAlongHeading);
// Define regular wave set
WaveSet1 = RegularWaveSet();
WaveSet1.add(RegularWave(270,26,WavePeriod(14),-60));
WaveSet1.add(RegularWave(180,15,WavePeriod(12),-60));
WaveSet1.add(RegularWave(90,28,WavePeriod(14),-60));
WaveSet1.add(RegularWave(0,27,WavePeriod(15),-60));
// Define wave load condition
Condition1 = DeterministicTime(Location1);
Condition1.waterSurface.regularWaveSet = WaveSet1;
Condition1.populate();
Condition1.addCalmSea();
Condition1.component(1).water.current(CurrentProfile1);
Condition1.component(1).waterSurface.waveModel(Stokes5());
Condition1.component(2).water.current(CurrentProfile1);
Condition1.component(2).waterSurface.waveModel(Stokes5());
Condition1.component(3).water.current(CurrentProfile1);
Condition1.component(3).waterSurface.waveModel(Stokes5());
Condition1.component(4).water.current(CurrentProfile1);
Condition1.component(4).waterSurface.waveModel(Stokes5());
// Define Morison coefficient and assign to jacket
MorisonConstant1 = MorisonCoefficients(0,0.7,0.7,0,2,2);
Jacket.morison = MorisonConstant1;
// Define marine growth and assign to jacket
MarineGrowthZLevel1 = MarineGrowthZLevelFunction();
MarineGrowthZLevel1.add(2,0.1,0.02,1.3);
MarineGrowthZLevel1.add(-24,0.05,0.02,1.3);
MarineGrowthZLevel1.add(-30,0,0.01,1.3);
MarineGrowthZLevel1.useInForceCalculations(true);
Jacket.marineGrowth = MarineGrowthZLevel1;
// Define legs as flooded
Flooding1 = Flooding(1);
Legs.flooding = Flooding1;

```

```

// Define analysis activity
Analysis1 = Analysis();
Analysis1.add(MeshActivity());
Analysis1.add(WaveLoadActivity(Condition1));
Analysis1.add(LinearAnalysis());
Analysis1.add(LoadResultsActivity());
// Wave load analysis activity
Analysis1.step(2).deterministicSeastates().populate();
Analysis1.step(2).deterministicSeastates().seastate(1).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(2).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(3).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(4).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(5).dataPhase(NoStretching, 0, 1, true, NoDesignLoads, 1, 1);
// Create load combinations
NorMaxSh = LoadCombination(Analysis1);
NorMaxSh.addCase(Analysis1.WLC(1), 1.6);
NorMaxSh.addCase(Analysis1.WLC(9), 1);
NorMaxSh.addCase(Gravity, 1.2);
NorMaxSh.addCase(WindN, 1.6);
NorMaxMo = LoadCombination(Analysis1);
NorMaxMo.addCase(Analysis1.WLC(2), 1.6);
NorMaxMo.addCase(Analysis1.WLC(9), 1);
NorMaxMo.addCase(Gravity, 1.2);
NorMaxMo.addCase(WindN, 1.6);
EasMaxSh = LoadCombination(Analysis1);
EasMaxSh.addCase(Analysis1.WLC(3), 1.6);
EasMaxSh.addCase(Analysis1.WLC(9), 1);
EasMaxSh.addCase(Gravity, 1.2);
EasMaxSh.addCase(WindE, 1.6);
EasMaxMo = LoadCombination(Analysis1);
EasMaxMo.addCase(Analysis1.WLC(4), 1.6);
EasMaxMo.addCase(Analysis1.WLC(9), 1);
EasMaxMo.addCase(Gravity, 1.2);
EasMaxMo.addCase(WindE, 1.6);
SouMaxSh = LoadCombination(Analysis1);
SouMaxSh.addCase(Analysis1.WLC(5), 1.6);
SouMaxSh.addCase(Analysis1.WLC(9), 1);
SouMaxSh.addCase(Gravity, 1.2);
SouMaxSh.addCase(WindS, 1.6);
SouMaxMo = LoadCombination(Analysis1);
SouMaxMo.addCase(Analysis1.WLC(6), 1.6);
SouMaxMo.addCase(Analysis1.WLC(9), 1);

```



```
SouMaxMo.addCase(Gravity,1.2);
SouMaxMo.addCase(WindS,1.6);
WesMaxSh = LoadCombination(Analysis1);
WesMaxSh.addCase(Analysis1.WLC(7),1.6);
WesMaxSh.addCase(Analysis1.WLC(9),1);
WesMaxSh.addCase(Gravity,1.2);
WesMaxSh.addCase(WindW,1.6);
WesMaxMo = LoadCombination(Analysis1);
WesMaxMo.addCase(Analysis1.WLC(8),1.6);
WesMaxMo.addCase(Analysis1.WLC(9),1);
WesMaxMo.addCase(Gravity,1.2);
WesMaxMo.addCase(WindW,1.6);
*/
```

## A 2 Fixed 4-Legged Jacket Input for Deterministic Fatigue Wave Analysis

The input below creates the 4-legged jacket shown in Figure A.1. The model is used in the tutorial example of Section 3.3. Note that the part of the input defining wave load data has been commented out (all lines between the line starting with ‘/\*’ and the last line ‘\*/’). Remove these two lines if you want to run through the complete example rather than working through the tutorial.

```
// Fixed 4 Legged jacket with risers and simplified topside, deterministic fatigue analysis
// Complete input except that definition of wave load data is commented out, search for '/*' to find where this data starts
// Created by Aanund B. Berdal, DNV Software
// Revised Sept. 15, 2005
GenieRules.Units.setDatabaseUnits("m", "kN", "delC");
GenieRules.Units.setInputUnit(Length, "m");
GenieRules.Units.setInputUnit(Force, "kN");
GenieRules.Units.setInputUnit(TempDiff, "delC");
GenieRules.Tolerances.useTolerantModelling = true;
// Sections and material
Pipe06 = PipeSection(0.6,0.01);
Pipe12 = PipeSection(1.2,.03);
Pipe16 = PipeSection(1.6,.03);
Pipe21 = PipeSection(2.1,.08);
Pipe22 = PipeSection(2.2,.08);
Pipe32 = PipeSection(3.2,.09);
Steel = Material(3.56E5,7.85,2.1E8,0.3,0.12E-04,0);
Steel.setDefault();
// Create legs
Pipe32.setDefault();
Leg1 = Beam(Point(-22.5 m,-17.5 m,0 m),Point(-10 m,-7.5 m,138 m));
Leg2 = Beam(Point(22.5 m,-17.5 m,0 m),Point(10 m,-7.5 m,138 m));
Leg3 = Beam(Point(22.5 m,17.5 m,0 m),Point(10 m,7.5 m,138 m));
Leg4 = Beam(Point(-22.5 m,17.5 m,0 m),Point(-10 m,7.5 m,138 m));
// Fix lower end of legs
Sp4 = SupportPoint(Point(-22.5 m,-17.5 m,0 m));
Sp5 = SupportPoint(Point(22.5 m,-17.5 m,0 m));
Sp6 = SupportPoint(Point(22.5 m,17.5 m,0 m));
Sp7 = SupportPoint(Point(-22.5 m,17.5 m,0 m));
// Create bracing between elevations 5 and 75
Pipe16.setDefault();
Br01 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,-17.13768116 m,5 m));
Br02 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br03 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
```

```

Br04 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,-17.13768116 m,5 m));
Br05 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br06 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
Br07 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));
Br08 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br09 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br10 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));
Br11 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br12 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br13 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br14 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br15 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br16 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br17 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br18 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br19 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br20 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br21 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br22 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br23 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(18.87681159 m,14.60144928 m,40 m));
Br24 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br25 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(15.70652174 m,12.06521739 m,75 m));
Br26 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-15.70652174 m,12.06521739 m,75 m));
// Create bracing between elevations 75 and 135
Pipe12.setDefault();
Br27 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br28 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br29 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br30 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br31 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br32 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br33 = Beam(Point(10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br34 = Beam(Point(-10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br35 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br36 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br37 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));
Br38 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br39 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br40 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));
Br41 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br42 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br43 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));

```

```

Br44 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br45 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br46 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br47 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br48 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br49 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br50 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br51 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-9.891304348 m,105 m));
Br52 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-7.717391304 m,135 m));
Br53 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,9.891304348 m,105 m));
Br54 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,7.717391304 m,135 m));
// Create horizontal bracing at elevations 105 and 135
Pipe16.setDefault();
Br55 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br56 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br57 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br58 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
// Change section for upper legs
Cone1 = ConeSection(1.0,true);
Leg1.divideSegmentAt(1,0.575848871);
Leg1.divideSegmentAt(2,0.05091188302);
Leg1.SetSegmentSection(3,Pipe22);
Leg1.SetSegmentSection(2,Cone1);
Leg2.divideSegmentAt(1,0.575848871);
Leg2.divideSegmentAt(2,0.05091188302);
Leg2.SetSegmentSection(3,Pipe22);
Leg2.SetSegmentSection(2,Cone1);
Leg3.divideSegmentAt(1,0.575848871);
Leg3.divideSegmentAt(2,0.05091188302);
Leg3.SetSegmentSection(3,Pipe22);
Leg3.SetSegmentSection(2,Cone1);
Leg4.divideSegmentAt(1,0.575848871);
Leg4.divideSegmentAt(2,0.05091188302);
Leg4.SetSegmentSection(3,Pipe22);
Leg4.SetSegmentSection(2,Cone1);
// Create short stubs at top of legs
Pipe22.setDefault();
Col1 = Beam(Point(-10 m,-7.5 m,138 m),Point(-10 m,-7.5 m,138 m + 5 m));
Col2 = Col1.copyTranslate(Vector3d(20 m,0 m,0 m));
Col3 = Col1.copyTranslate(Vector3d(0 m,15 m,0 m));
Col4 = Col2.copyTranslate(Vector3d(0 m,15 m,0 m));
// Create simplified topside

```

```

Th2 = Thickness(0.02);
Th4 = Thickness(0.04);
Th6 = Thickness(0.06);
Th6.setDefault();
DeckLow = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,7.5 m,143 m),Point(-22 m,7.5 m,143 m));
DeckMid = DeckLow.copyTranslate(Vector3d(0 m,0 m,9 m));
DeckUpp = DeckLow.copyTranslate(Vector3d(0 m,0 m,17 m));
WallyZ1 = Plate(Point(-22 m,-7.5 m,143 m),Point(-22 m,7.5 m,143 m),Point(-22 m,7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallyZ2 = WallyZ1.copyTranslate(Vector3d(12 m,0 m,0 m));
WallyZ3 = WallyZ1.copyTranslate(Vector3d(32 m,0 m,0 m));
WallyZ4 = WallyZ1.copyTranslate(Vector3d(48 m,0 m,0 m));
WallXZ2 = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,-7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallXZ2.moveTranslate(Vector3d(0 m,10.5 m,0 m),geUNCONNECTED);
WallXZ1 = Plate(Point(-10 m,-7.5 m,143 m),Point(10 m,-7.5 m,143 m),Point(10 m,-7.5 m,160 m),Point(-10 m,-7.5 m,160 m));
WallXZ3 = WallXZ1.copyTranslate(Vector3d(0 m,15 m,0 m));
// Add mass points to upper deck of topside
Mass1 = PointMass(Point(-22 m,-7.5 m,160 m),600);
Mass2 = PointMass(Point(-22 m,3 m,160 m),600);
Mass3 = PointMass(Point(-22 m,7.5 m,160 m),600);
Mass4 = PointMass(Point(-10 m,-7.5 m,160 m),600);
Mass5 = PointMass(Point(-10 m,3 m,160 m),600);
Mass6 = PointMass(Point(-10 m,7.5 m,160 m),600);
Mass7 = PointMass(Point(10 m,-7.5 m,160 m),600);
Mass8 = PointMass(Point(10 m,3 m,160 m),600);
Mass9 = PointMass(Point(10 m,7.5 m,160 m),600);
Mass10 = PointMass(Point(26 m,-7.5 m,160 m),600);
Mass11 = PointMass(Point(26 m,3 m,160 m),600);
Mass12 = PointMass(Point(26 m,7.5 m,160 m),600);
// Create conductor supports and non-structural conductors
Pipe16.setDefault();
Cndsupp1 = Beam(Point(-3.992957746 m,3 m,135 m),Point(3.992957746 m,3 m,135 m));
Cndsupp2 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-30 m));
Cndsupp3 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-60 m));
Cndsupp4 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-95 m));
Cndsupp5 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-130 m));
Pipe06.setDefault();
Cnd1 = Beam(Point(-2 m,3 m,0 m),Point(-2 m,3 m,143 m));
Cnd2 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*1);
Cnd3 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*2);
Non = BeamTypeNonstruct();
Cnd1.beamType = Non;
Cnd2.beamType = Non;

```

```
Cnd3.beamType = Non;
// Fix lower ends of non-structural conductors that have no stiffness
Sp1 = SupportPoint(Point(-2 m,3 m,0 m));
Sp2 = SupportPoint(Point(0 m,3 m,0 m));
Sp3 = SupportPoint(Point(2 m,3 m,0 m));
// Create set named Jacket including all beams above mudline and up to lower deck of topside
Jacket = Set();
Jacket.add(Leg1);
Jacket.add(Leg2);
Jacket.add(Leg3);
Jacket.add(Leg4);
Jacket.add(Br01);
Jacket.add(Br02);
Jacket.add(Br03);
Jacket.add(Br04);
Jacket.add(Br05);
Jacket.add(Br06);
Jacket.add(Br07);
Jacket.add(Br08);
Jacket.add(Br09);
Jacket.add(Br10);
Jacket.add(Br11);
Jacket.add(Br12);
Jacket.add(Br13);
Jacket.add(Br14);
Jacket.add(Br15);
Jacket.add(Br16);
Jacket.add(Br17);
Jacket.add(Br18);
Jacket.add(Br19);
Jacket.add(Br20);
Jacket.add(Br21);
Jacket.add(Br22);
Jacket.add(Br23);
Jacket.add(Br24);
Jacket.add(Br25);
Jacket.add(Br26);
Jacket.add(Br27);
Jacket.add(Br28);
Jacket.add(Br29);
Jacket.add(Br30);
Jacket.add(Br31);
```

```
Jacket.add(Br32);
Jacket.add(Br33);
Jacket.add(Br34);
Jacket.add(Br35);
Jacket.add(Br36);
Jacket.add(Br37);
Jacket.add(Br38);
Jacket.add(Br39);
Jacket.add(Br40);
Jacket.add(Br41);
Jacket.add(Br42);
Jacket.add(Br43);
Jacket.add(Br44);
Jacket.add(Br45);
Jacket.add(Br46);
Jacket.add(Br47);
Jacket.add(Br48);
Jacket.add(Br49);
Jacket.add(Br50);
Jacket.add(Br51);
Jacket.add(Br52);
Jacket.add(Br53);
Jacket.add(Br54);
Jacket.add(Br55);
Jacket.add(Br56);
Jacket.add(Br57);
Jacket.add(Br58);
Jacket.add(Col1);
Jacket.add(Col2);
Jacket.add(Col3);
Jacket.add(Col4);
Jacket.add(Cndsupp1);
Jacket.add(Cndsupp2);
Jacket.add(Cndsupp3);
Jacket.add(Cndsupp4);
Jacket.add(Cndsupp5);
Jacket.add(Cnd2);
Jacket.add(Cnd1);
Jacket.add(Cnd3);
// Create set named Legs including the four legs
Legs = Set();
Legs.add(Leg2);
```

```

Legs.add(Leg1);
Legs.add(Leg3);
Legs.add(Leg4);
// Create set named Topside including decks and walls of topside
Topside = Set();
Topside.add(DeckUpp);
Topside.add(DeckMid);
Topside.add(DeckLow);
Topside.add(WallYZ1);
Topside.add(WallYZ2);
Topside.add(WallYZ3);
Topside.add(WallYZ4);
Topside.add(WallXZ1);
Topside.add(WallXZ2);
Topside.add(WallXZ3);
// Define mesh density for topside, feature edge used to control mesh
Mesh1 = MeshDensity(20);
Topside.meshDensity = Mesh1;
FEdge1 = FeatureEdge(Point(-2 m,-7.5 m,143 m),Point(-2 m,7.5 m,143 m));
FEdge2 = FeatureEdge(Point(-2 m,-7.5 m,152 m),Point(-2 m,7.5 m,152 m));
FEdge3 = FeatureEdge(Point(-2 m,-7.5 m,160 m),Point(-2 m,7.5 m,160 m));
FEdge4 = FeatureEdge(Point(-2 m,3 m,143 m),Point(-2 m,3 m,160 m));
FEdge5 = FeatureEdge(Point(0 m,-7.5 m,143 m),Point(0 m,7.5 m,143 m));
FEdge6 = FeatureEdge(Point(0 m,-7.5 m,152 m),Point(0 m,7.5 m,152 m));
FEdge7 = FeatureEdge(Point(0 m,-7.5 m,160 m),Point(0 m,7.5 m,160 m));
FEdge8 = FeatureEdge(Point(0 m,3 m,143 m),Point(0 m,3 m,160 m));
FEdge9 = FeatureEdge(Point(2 m,-7.5 m,143 m),Point(2 m,7.5 m,143 m));
FEdge10 = FeatureEdge(Point(2 m,-7.5 m,152 m),Point(2 m,7.5 m,152 m));
FEdge11 = FeatureEdge(Point(2 m,-7.5 m,160 m),Point(2 m,7.5 m,160 m));
FEdge12 = FeatureEdge(Point(2 m,3 m,143 m),Point(2 m,3 m,160 m));
/* The rest of the input is commented out
// Here hydro properties starts
// Define location
Location1 = Location();
Location1.gravity = 9.80665 m/s^2;
Location1.air.density = 0.001226 tonne/m^3;
Location1.air.kinematicViscosity = 1.492e-008 kN*s/m^2;
Location1.water.density = 1.025 tonne/m^3;
Location1.water.kinematicViscosity = 1.981e-009 kN*s/m^2;
Location1.waterSurface.z = 124 m;
Location1.seabed.normaldirection = Vector3d(0 m,0 m,1 m);
Location1.seabed.z = 0 m;

```



```

// Define regular wave set
WaveSet1 = RegularWaveSet();
WaveSet1.add(RegularWave(0 deg,4 m,WavePeriod(6.5 s),0 deg));
WaveSet1.add(RegularWave(0 deg,8 m,WavePeriod(9.2 s),0 deg));
WaveSet1.add(RegularWave(0 deg,12 m,WavePeriod(11.2 s),0 deg));
WaveSet1.add(RegularWave(0 deg,15 m,WavePeriod(12.6 s),0 deg));
WaveSet1.add(RegularWave(45 deg,4 m,WavePeriod(6.5 s),0 deg));
WaveSet1.add(RegularWave(45 deg,8 m,WavePeriod(9.2 s),0 deg));
WaveSet1.add(RegularWave(45 deg,12 m,WavePeriod(11.2 s),0 deg));
WaveSet1.add(RegularWave(45 deg,15 m,WavePeriod(12.6 s),0 deg));
WaveSet1.add(RegularWave(90 deg,4 m,WavePeriod(6.5 s),0 deg));
WaveSet1.add(RegularWave(90 deg,8 m,WavePeriod(9.2 s),0 deg));
WaveSet1.add(RegularWave(90 deg,12 m,WavePeriod(11.2 s),0 deg));
WaveSet1.add(RegularWave(90 deg,15 m,WavePeriod(12.6 s),0 deg));
WaveSet1.add(RegularWave(135 deg,4 m,WavePeriod(6.5 s),0 deg));
WaveSet1.add(RegularWave(135 deg,8 m,WavePeriod(9.2 s),0 deg));
WaveSet1.add(RegularWave(135 deg,12 m,WavePeriod(11.2 s),0 deg));
WaveSet1.add(RegularWave(135 deg,15 m,WavePeriod(12.6 s),0 deg));
// Define wave load condition
Condition1 = DeterministicTime(Location1);
Condition1.waterSurface.regularWaveSet = WaveSet1;
Condition1.populate();
Condition1.component(1).waterSurface.waveModel(Airy());
Condition1.component(2).waterSurface.waveModel(Airy());
Condition1.component(3).waterSurface.waveModel(Airy());
Condition1.component(4).waterSurface.waveModel(Airy());
Condition1.component(5).waterSurface.waveModel(Airy());
Condition1.component(6).waterSurface.waveModel(Airy());
Condition1.component(7).waterSurface.waveModel(Airy());
Condition1.component(8).waterSurface.waveModel(Airy());
Condition1.component(9).waterSurface.waveModel(Airy());
Condition1.component(10).waterSurface.waveModel(Airy());
Condition1.component(11).waterSurface.waveModel(Airy());
Condition1.component(12).waterSurface.waveModel(Airy());
Condition1.component(13).waterSurface.waveModel(Airy());
Condition1.component(14).waterSurface.waveModel(Airy());
Condition1.component(15).waterSurface.waveModel(Airy());
Condition1.component(16).waterSurface.waveModel(Airy());
// Define Morison coefficient and assign to jacket
MorisonConstant1 = MorisonCoefficients(0,0.7,0.7,0,2,2);
Jacket.morison = MorisonConstant1;
// Define marine growth and assign to jacket

```

```
MarineGrowthZLevel1 = MarineGrowthZLevelFunction();
MarineGrowthZLevel1.add(2,0.1,0.02,1.3);
MarineGrowthZLevel1.add(-24,0.05,0.02,1.3);
MarineGrowthZLevel1.add(-30,0,0.01,1.3);
MarineGrowthZLevel1.useInForceCalculations(true);
Jacket.marineGrowth = MarineGrowthZLevel1;
// Define legs as flooded
Flooding1 = Flooding(1);
Legs.flooding = Flooding1;
// Define element refinement
ElementRefinement1 = ElementRefinement(5, true, 1 m, 2 m);
ElementRefinement2 = ElementRefinement(15, true, 1 m, 2 m);
Br27.elementRefinement = ElementRefinement2;
Br28.elementRefinement = ElementRefinement2;
Br29.elementRefinement = ElementRefinement2;
Br30.elementRefinement = ElementRefinement2;
Br31.elementRefinement = ElementRefinement2;
Br32.elementRefinement = ElementRefinement2;
Br33.elementRefinement = ElementRefinement2;
Br34.elementRefinement = ElementRefinement2;
Br43.elementRefinement = ElementRefinement2;
Br45.elementRefinement = ElementRefinement2;
Br44.elementRefinement = ElementRefinement2;
Br46.elementRefinement = ElementRefinement2;
Br47.elementRefinement = ElementRefinement2;
Br48.elementRefinement = ElementRefinement2;
Br50.elementRefinement = ElementRefinement2;
Br49.elementRefinement = ElementRefinement2;
Br51.elementRefinement = ElementRefinement2;
Br52.elementRefinement = ElementRefinement2;
Br53.elementRefinement = ElementRefinement2;
Br55.elementRefinement = ElementRefinement2;
Br54.elementRefinement = ElementRefinement2;
Br56.elementRefinement = ElementRefinement2;
Br57.elementRefinement = ElementRefinement2;
Br58.elementRefinement = ElementRefinement2;
Leg1.elementRefinement = ElementRefinement1;
Leg2.elementRefinement = ElementRefinement1;
Leg3.elementRefinement = ElementRefinement1;
Leg4.elementRefinement = ElementRefinement1;
Br01.elementRefinement = ElementRefinement1;
Br02.elementRefinement = ElementRefinement1;
```

```

Br03.elementRefinement = ElementRefinement1;
Br05.elementRefinement = ElementRefinement1;
Br04.elementRefinement = ElementRefinement1;
Br06.elementRefinement = ElementRefinement1;
Br07.elementRefinement = ElementRefinement1;
Br08.elementRefinement = ElementRefinement1;
Br10.elementRefinement = ElementRefinement1;
Br09.elementRefinement = ElementRefinement1;
Br11.elementRefinement = ElementRefinement1;
Br12.elementRefinement = ElementRefinement1;
Br13.elementRefinement = ElementRefinement1;
Br15.elementRefinement = ElementRefinement1;
Br14.elementRefinement = ElementRefinement1;
Br16.elementRefinement = ElementRefinement1;
Br17.elementRefinement = ElementRefinement1;
Br18.elementRefinement = ElementRefinement1;
Br20.elementRefinement = ElementRefinement1;
Br19.elementRefinement = ElementRefinement1;
Br21.elementRefinement = ElementRefinement1;
Br22.elementRefinement = ElementRefinement1;
Br23.elementRefinement = ElementRefinement1;
Br25.elementRefinement = ElementRefinement1;
Br24.elementRefinement = ElementRefinement1;
Br26.elementRefinement = ElementRefinement1;
Br35.elementRefinement = ElementRefinement1;
Br36.elementRefinement = ElementRefinement1;
Br37.elementRefinement = ElementRefinement1;
Br40.elementRefinement = ElementRefinement1;
Br38.elementRefinement = ElementRefinement1;
Br39.elementRefinement = ElementRefinement1;
Br41.elementRefinement = ElementRefinement1;
Br42.elementRefinement = ElementRefinement1;
// Define analysis activity
Analysis1 = Analysis();
Analysis1.add(MeshActivity());
Analysis1.add(WaveLoadActivity(Condition1));
Analysis1.add(LinearAnalysis());
Analysis1.add(LoadResultsActivity());
// Wave load analysis activity
Analysis1.step(2).deterministicSeastates().populate();
Analysis1.step(2).deterministicSeastates().seastate(1).stretchType(WheelerStretching);
Analysis1.step(2).deterministicSeastates().seastate(1).stepLengthPhase(30 deg);

```

[illegible]

[illegible]

```

Analysis1.step(2).deterministicSeastates().seastate(15).numberOfSteps(12);
Analysis1.step(2).deterministicSeastates().seastate(15).buoyancy(false);
Analysis1.step(2).deterministicSeastates().seastate(15).designloadType(NoDesignLoads);
Analysis1.step(2).deterministicSeastates().seastate(15).waveKinematicsFactor(1);
Analysis1.step(2).deterministicSeastates().seastate(16).stretchType(WheelerStretching);
Analysis1.step(2).deterministicSeastates().seastate(16).stepLengthPhase(30 deg);
Analysis1.step(2).deterministicSeastates().seastate(16).numberOfSteps(12);
Analysis1.step(2).deterministicSeastates().seastate(16).buoyancy(false);
Analysis1.step(2).deterministicSeastates().seastate(16).designloadType(NoDesignLoads);
Analysis1.step(2).deterministicSeastates().seastate(16).waveKinematicsFactor(1);
*/

```

## A 3 4-Legged Jacket Input for Structure-Pile-Soil Analysis

The input below creates the 4-legged jacket shown in Figure A.1. The model is used in the tutorial example of Section 3.6. Note that the part of the input defining pile, soil and other data has been commented out (all lines between the line starting with ‘/\*’ and the last line ‘\*/’). Remove these two lines if you want to run through the complete example rather than working through the tutorial.

```

// 4 Legged jacket with piles, risers and simplified topside
// Complete input except that definition of soil, piles and more is commented out, search for '/*' to find where this data starts
// Created by Aanund B. Berdal, DNV Software
// Revised Sept. 15, 2005
GenieRules.Units.setDatabaseUnits("m", "kN", "delC");
GenieRules.Units.setInputUnit(Length, "m");
GenieRules.Units.setInputUnit(Force, "kN");
GenieRules.Units.setInputUnit(TempDiff, "delC");
GenieRules.Tolerances.useTolerantModelling = true;
// Sections and material
Pipe06 = PipeSection(0.6,0.01);
Pipe12 = PipeSection(1.2,.03);
Pipe16 = PipeSection(1.6,.03);
Pipe21 = PipeSection(2.1,.08);
Pipe22 = PipeSection(2.2,.08);
Pipe32 = PipeSection(3.2,.09);
Steel = Material(3.56E5,7.85,2.1E8,0.3,0.12E-04,0);
Steel.setDefault();
// Create legs
Pipe32.setDefault();
Leg1 = Beam(Point(-22.5 m,-17.5 m,0 m),Point(-10 m,-7.5 m,138 m));
Leg2 = Beam(Point(22.5 m,-17.5 m,0 m),Point(10 m,-7.5 m,138 m));
Leg3 = Beam(Point(22.5 m,17.5 m,0 m),Point(10 m,7.5 m,138 m));

```

```

Leg4 = Beam(Point(-22.5 m,17.5 m,0 m),Point(-10 m,7.5 m,138 m));
// Create bracing between elevations 5 and 75
Pipe16.setDefault();
Br01 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,-17.13768116 m,5 m));
Br02 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br03 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
Br04 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-22.04710145 m,-17.13768116 m,5 m));
Br05 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(22.04710145 m,17.13768116 m,5 m));
Br06 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-22.04710145 m,17.13768116 m,5 m));
Br07 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));
Br08 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br09 = Beam(Point(22.04710145 m,-17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br10 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,-14.60144928 m,40 m));
Br11 = Beam(Point(22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br12 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(18.87681159 m,14.60144928 m,40 m));
Br13 = Beam(Point(-22.04710145 m,17.13768116 m,5 m),Point(-18.87681159 m,-14.60144928 m,40 m));
Br14 = Beam(Point(-22.04710145 m,-17.13768116 m,5 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br15 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br16 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br17 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br18 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,-12.06521739 m,75 m));
Br19 = Beam(Point(18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br20 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(15.70652174 m,12.06521739 m,75 m));
Br21 = Beam(Point(-18.87681159 m,14.60144928 m,40 m),Point(-15.70652174 m,-12.06521739 m,75 m));
Br22 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(-15.70652174 m,12.06521739 m,75 m));
Br23 = Beam(Point(-18.87681159 m,-14.60144928 m,40 m),Point(18.87681159 m,14.60144928 m,40 m));
Br24 = Beam(Point(18.87681159 m,-14.60144928 m,40 m),Point(-18.87681159 m,14.60144928 m,40 m));
Br25 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(15.70652174 m,12.06521739 m,75 m));
Br26 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-15.70652174 m,12.06521739 m,75 m));
// Create bracing between elevations 75 and 135
Pipe12.setDefault();
Br27 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br28 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br29 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br30 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br31 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br32 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br33 = Beam(Point(10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br34 = Beam(Point(-10.27173913 m,7.717391304 m,135 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br35 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br36 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br37 = Beam(Point(15.70652174 m,-12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));

```

```

Br38 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,-9.891304348 m,105 m));
Br39 = Beam(Point(15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br40 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(12.98913043 m,9.891304348 m,105 m));
Br41 = Beam(Point(-15.70652174 m,12.06521739 m,75 m),Point(-12.98913043 m,-9.891304348 m,105 m));
Br42 = Beam(Point(-15.70652174 m,-12.06521739 m,75 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br43 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br44 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br45 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br46 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,-7.717391304 m,135 m));
Br47 = Beam(Point(12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br48 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(10.27173913 m,7.717391304 m,135 m));
Br49 = Beam(Point(-12.98913043 m,9.891304348 m,105 m),Point(-10.27173913 m,-7.717391304 m,135 m));
Br50 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(-10.27173913 m,7.717391304 m,135 m));
Br51 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-9.891304348 m,105 m));
Br52 = Beam(Point(0 m,-8.677366924 m,121.7523364 m),Point(0 m,-7.717391304 m,135 m));
Br53 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,9.891304348 m,105 m));
Br54 = Beam(Point(0 m,8.677366924 m,121.7523364 m),Point(0 m,7.717391304 m,135 m));
// Create horizontal bracing at elevations 105 and 135
Pipe16.setDefault();
Br55 = Beam(Point(-12.98913043 m,-9.891304348 m,105 m),Point(12.98913043 m,9.891304348 m,105 m));
Br56 = Beam(Point(12.98913043 m,-9.891304348 m,105 m),Point(-12.98913043 m,9.891304348 m,105 m));
Br57 = Beam(Point(-10.27173913 m,-7.717391304 m,135 m),Point(10.27173913 m,7.717391304 m,135 m));
Br58 = Beam(Point(10.27173913 m,-7.717391304 m,135 m),Point(-10.27173913 m,7.717391304 m,135 m));
// Change section for upper legs
Conel = ConeSection(1.0,true);
Leg1.divideSegmentAt(1,0.575848871);
Leg1.divideSegmentAt(2,0.05091188302);
Leg1.SetSegmentSection(3,Pipe22);
Leg1.SetSegmentSection(2,Conel);
Leg2.divideSegmentAt(1,0.575848871);
Leg2.divideSegmentAt(2,0.05091188302);
Leg2.SetSegmentSection(3,Pipe22);
Leg2.SetSegmentSection(2,Conel);
Leg3.divideSegmentAt(1,0.575848871);
Leg3.divideSegmentAt(2,0.05091188302);
Leg3.SetSegmentSection(3,Pipe22);
Leg3.SetSegmentSection(2,Conel);
Leg4.divideSegmentAt(1,0.575848871);
Leg4.divideSegmentAt(2,0.05091188302);
Leg4.SetSegmentSection(3,Pipe22);
Leg4.SetSegmentSection(2,Conel);
// Create short stubs at top of legs

```



```

Pipe22.setDefault();
Col1 = Beam(Point(-10 m,-7.5 m,138 m),Point(-10 m,-7.5 m,138 m + 5 m));
Col2 = Col1.copyTranslate(Vector3d(20 m,0 m,0 m));
Col3 = Col1.copyTranslate(Vector3d(0 m,15 m,0 m));
Col4 = Col2.copyTranslate(Vector3d(0 m,15 m,0 m));
// Create simplified topside
Th2 = Thickness(0.02);
Th4 = Thickness(0.04);
Th6 = Thickness(0.06);
Th6.setDefault();
DeckLow = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,7.5 m,143 m),Point(-22 m,7.5 m,143 m));
DeckMid = DeckLow.copyTranslate(Vector3d(0 m,0 m,9 m));
DeckUpp = DeckLow.copyTranslate(Vector3d(0 m,0 m,17 m));
WallYZ1 = Plate(Point(-22 m,-7.5 m,143 m),Point(-22 m,7.5 m,143 m),Point(-22 m,7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallYZ2 = WallYZ1.copyTranslate(Vector3d(12 m,0 m,0 m));
WallYZ3 = WallYZ1.copyTranslate(Vector3d(32 m,0 m,0 m));
WallYZ4 = WallYZ1.copyTranslate(Vector3d(48 m,0 m,0 m));
WallXZ2 = Plate(Point(-22 m,-7.5 m,143 m),Point(26 m,-7.5 m,143 m),Point(26 m,-7.5 m,160 m),Point(-22 m,-7.5 m,160 m));
WallXZ2.moveTranslate(Vector3d(0 m,10.5 m,0 m),geUNCONNECTED);
WallXZ1 = Plate(Point(-10 m,-7.5 m,143 m),Point(10 m,-7.5 m,143 m),Point(10 m,-7.5 m,160 m),Point(-10 m,-7.5 m,160 m));
WallXZ3 = WallXZ1.copyTranslate(Vector3d(0 m,15 m,0 m));
// Add mass points to upper deck of topside
Mass1 = PointMass(Point(-22 m,-7.5 m,160 m),600);
Mass2 = PointMass(Point(-22 m,3 m,160 m),600);
Mass3 = PointMass(Point(-22 m,7.5 m,160 m),600);
Mass4 = PointMass(Point(-10 m,-7.5 m,160 m),600);
Mass5 = PointMass(Point(-10 m,3 m,160 m),600);
Mass6 = PointMass(Point(-10 m,7.5 m,160 m),600);
Mass7 = PointMass(Point(10 m,-7.5 m,160 m),600);
Mass8 = PointMass(Point(10 m,3 m,160 m),600);
Mass9 = PointMass(Point(10 m,7.5 m,160 m),600);
Mass10 = PointMass(Point(26 m,-7.5 m,160 m),600);
Mass11 = PointMass(Point(26 m,3 m,160 m),600);
Mass12 = PointMass(Point(26 m,7.5 m,160 m),600);
// Create conductor supports and non-structural conductors
Pipe16.setDefault();
Cndsupp1 = Beam(Point(-3.992957746 m,3 m,135 m),Point(3.992957746 m,3 m,135 m));
Cndsupp2 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-30 m));
Cndsupp3 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-60 m));
Cndsupp4 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-95 m));
Cndsupp5 = Cndsupp1.copyTranslate(Vector3d(0 m,0 m,-130 m));
Pipe06.setDefault();

```

```
Cnd1 = Beam(Point(-2 m,3 m,0 m),Point(-2 m,3 m,143 m));
Cnd2 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*1);
Cnd3 = Cnd1.copyTranslate(Vector3d(2 m,0 m,0 m)*2);
Non = BeamTypeNonstruct();
Cnd1.beamType = Non;
Cnd2.beamType = Non;
Cnd3.beamType = Non;
// Fix lower ends of non-structural conductors that have no stiffness
Sp1 = SupportPoint(Point(-2 m,3 m,0 m));
Sp2 = SupportPoint(Point(0 m,3 m,0 m));
Sp3 = SupportPoint(Point(2 m,3 m,0 m));
// Create set named Jacket including all beams above mudline and up to lower deck of topside
Jacket = Set();
Jacket.add(Leg1);
Jacket.add(Leg2);
Jacket.add(Leg3);
Jacket.add(Leg4);
Jacket.add(Br01);
Jacket.add(Br02);
Jacket.add(Br03);
Jacket.add(Br04);
Jacket.add(Br05);
Jacket.add(Br06);
Jacket.add(Br07);
Jacket.add(Br08);
Jacket.add(Br09);
Jacket.add(Br10);
Jacket.add(Br11);
Jacket.add(Br12);
Jacket.add(Br13);
Jacket.add(Br14);
Jacket.add(Br15);
Jacket.add(Br16);
Jacket.add(Br17);
Jacket.add(Br18);
Jacket.add(Br19);
Jacket.add(Br20);
Jacket.add(Br21);
Jacket.add(Br22);
Jacket.add(Br23);
Jacket.add(Br24);
Jacket.add(Br25);
```

```
Jacket.add(Br26);
Jacket.add(Br27);
Jacket.add(Br28);
Jacket.add(Br29);
Jacket.add(Br30);
Jacket.add(Br31);
Jacket.add(Br32);
Jacket.add(Br33);
Jacket.add(Br34);
Jacket.add(Br35);
Jacket.add(Br36);
Jacket.add(Br37);
Jacket.add(Br38);
Jacket.add(Br39);
Jacket.add(Br40);
Jacket.add(Br41);
Jacket.add(Br42);
Jacket.add(Br43);
Jacket.add(Br44);
Jacket.add(Br45);
Jacket.add(Br46);
Jacket.add(Br47);
Jacket.add(Br48);
Jacket.add(Br49);
Jacket.add(Br50);
Jacket.add(Br51);
Jacket.add(Br52);
Jacket.add(Br53);
Jacket.add(Br54);
Jacket.add(Br55);
Jacket.add(Br56);
Jacket.add(Br57);
Jacket.add(Br58);
Jacket.add(Col1);
Jacket.add(Col2);
Jacket.add(Col3);
Jacket.add(Col4);
Jacket.add(Cndsupp1);
Jacket.add(Cndsupp2);
Jacket.add(Cndsupp3);
Jacket.add(Cndsupp4);
Jacket.add(Cndsupp5);
```

```

Jacket.add(Cnd2);
Jacket.add(Cnd1);
Jacket.add(Cnd3);
// Create set named Legs including the four legs
Legs = Set();
Legs.add(Leg2);
Legs.add(Leg1);
Legs.add(Leg3);
Legs.add(Leg4);
// Create set named Topside including decks and walls of topside
Topside = Set();
Topside.add(DeckUpp);
Topside.add(DeckMid);
Topside.add(DeckLow);
Topside.add(WallYZ1);
Topside.add(WallYZ2);
Topside.add(WallYZ3);
Topside.add(WallYZ4);
Topside.add(WallXZ1);
Topside.add(WallXZ2);
Topside.add(WallXZ3);
// Define mesh density for topside, feature edge used to control mesh
Mesh1 = MeshDensity(20);
Topside.meshDensity = Mesh1;
FEdge1 = FeatureEdge(Point(-2 m,-7.5 m,143 m),Point(-2 m,7.5 m,143 m));
FEdge2 = FeatureEdge(Point(-2 m,-7.5 m,152 m),Point(-2 m,7.5 m,152 m));
FEdge3 = FeatureEdge(Point(-2 m,-7.5 m,160 m),Point(-2 m,7.5 m,160 m));
FEdge4 = FeatureEdge(Point(-2 m,3 m,143 m),Point(-2 m,3 m,160 m));
FEdge5 = FeatureEdge(Point(0 m,-7.5 m,143 m),Point(0 m,7.5 m,143 m));
FEdge6 = FeatureEdge(Point(0 m,-7.5 m,152 m),Point(0 m,7.5 m,152 m));
FEdge7 = FeatureEdge(Point(0 m,-7.5 m,160 m),Point(0 m,7.5 m,160 m));
FEdge8 = FeatureEdge(Point(0 m,3 m,143 m),Point(0 m,3 m,160 m));
FEdge9 = FeatureEdge(Point(2 m,-7.5 m,143 m),Point(2 m,7.5 m,143 m));
FEdge10 = FeatureEdge(Point(2 m,-7.5 m,152 m),Point(2 m,7.5 m,152 m));
FEdge11 = FeatureEdge(Point(2 m,-7.5 m,160 m),Point(2 m,7.5 m,160 m));
FEdge12 = FeatureEdge(Point(2 m,3 m,143 m),Point(2 m,3 m,160 m));
// Create gravity and wind load cases
Gravity = LoadCase();
Gravity.setAcceleration(Vector3d(0 m/s^2,0 m/s^2,-9.80665 m/s^2));
Gravity.includeSelfWeight();
WindN = LoadCase();
Wind1 = FlatSurfacePressure(WindN,5 KPa,Point(-22,3,143),Point(26,3,143),Point(26,3,160),Point(-22,3,160));

```

```

WindE = LoadCase();
Wind2 = FlatSurfacePressure(WindE, 7 KPa, Point(26, -7.5, 143), Point(26, -7.5, 160), Point(26, 7.5, 160), Point(26, 7.5, 143));
WindS = LoadCase();
Wind3 = FlatSurfacePressure(WindS, 6 KPa, Point(-22, 3, 143), Point(-22, 3, 160), Point(26, 3, 160), Point(26, 3, 143));
WindW = LoadCase();
Wind4 = FlatSurfacePressure(WindW, 9 KPa, Point(-22, 7.5, 143), Point(-22, 7.5, 160), Point(-22, -7.5, 160), Point(-22, -7.5, 143));
// Current profile
CProfile1_Elevations = Array(30, 0, -44, -124);
CProfile1_Directions = Array(0 deg, 0 deg, 0 deg, 0 deg);
CProfile1_Velocities = Array(1.1, 1.1, 0.5, 0);
CurrentProfile1 = CurrentProfileRelDir(CProfile1_Elevations, CProfile1_Directions, CProfile1_Velocities, dtAlongHeading);
// Wave set
WaveSet1 = RegularWaveSet();
WaveSet1.add(RegularWave(270, 26, WavePeriod(14), -60));
WaveSet1.add(RegularWave(180, 15, WavePeriod(12), -60));
WaveSet1.add(RegularWave(90, 28, WavePeriod(14), -60));
WaveSet1.add(RegularWave(0, 27, WavePeriod(15), -60));
// Morison coefficient assigned to set named Jacket
MorisonConstant1 = MorisonCoefficients(0, 0.7, 0.7, 0, 2, 2);
Jacket.morison = MorisonConstant1;
// Marine growth assigned to set named Jacket
MarineGrowthZLevel1 = MarineGrowthZLevelFunction();
MarineGrowthZLevel1.add(0, 0.1, 0, 1);
MarineGrowthZLevel1.add(-24, 0.1, 0, 1);
MarineGrowthZLevel1.add(-124, 0.01, 0, 1);
MarineGrowthZLevel1.useInForceCalculations(true);
Jacket.marineGrowth = MarineGrowthZLevel1;
// Flooding assigned to set named Legs
Flooding1 = Flooding(1);
Legs.flooding = Flooding1;
/* The rest of the input is commented out
// Pile characteristics
PileType1 = PileCharacteristics(1 tonne/m^3, tcInfiniteBelowZLevel, -15 m);
// Create piles
Pipe21.setDefault();
Pile1 = Pile(Point(-22.5 m, -17.5 m, 0 m), Point(-29.29347826 m, -22.93478261 m, -75 m));
Pile1.pileCharacteristics = PileType1;
Pile2 = Pile(Point(22.5 m, -17.5 m, 0 m), Point(29.29347826 m, -22.93478261 m, -75 m));
Pile2.pileCharacteristics = PileType1;
Pile3 = Pile(Point(22.5 m, 17.5 m, 0 m), Point(29.29347826 m, 22.93478261 m, -75 m));
Pile3.pileCharacteristics = PileType1;
Pile4 = Pile(Point(-22.5 m, 17.5 m, 0 m), Point(-29.29347826 m, 22.93478261 m, -75 m));

```

```

Pile4.pileCharacteristics = PileType1;
// Seabed deltas
Scour1 = Scour(0.5 m, 1 m, 20 deg);
// Soil types
Soil1 = Sand(false, 1, 1.99 tonne/m^3, 40 deg);
Soil2 = Sand(false, 1, 1.99 tonne/m^3, 36 deg);
Soil3 = Clay(false, 1, 1.94 tonne/m^3, 0.01, 200 KPa, 0 m, 100 KPa, -100 m);
Soil3.apiJFactor = 0.5;
Soil4 = Clay(false, 1, 1.94 tonne/m^3, 0.01, 300 KPa, 0 m, 130 KPa, -100 m);
Soil4.apiJFactor = 0.5;
Soil5 = Sand(false, 1, 2.04 tonne/m^3, 37 deg);
// Soil data and soil curves
SoilCurves1 = SoilCurves(pyAPI1987, tzAPI1993, qzAPI1993);
SoilData1 = SoilData(-1 KPa, 0.5, 5 KPa, 3 KPa, 0.01, 0 KPa, 0.05);
SoilData2 = SoilData(-1 KPa, 0.5, 15 KPa, 11 KPa, 0.01, 0 KPa, 0.05);
SoilData3 = SoilData(-1 KPa, 0.5, 45 KPa, 45 KPa, 0.01, 0 KPa, 0.05);
SoilData4 = SoilData(-1 KPa, 0.5, 200 KPa, 200 KPa, 0.01, 0 KPa, 0.05);
SoilData5 = SoilData(-1 KPa, 0.5, 250 KPa, 250 KPa, 0.01, 0 KPa, 0.05);
SoilData6 = SoilData(-1 KPa, 0.5, 120 KPa, 120 KPa, 0.01, 30000 KPa, 0.05);
// Location
Location1 = Location(124 m, 0 m);
Location1.gravity = 9.80665 m/s^2;
Location1.air().density = 0.001226 tonne/m^3;
Location1.air().kinematicViscosity = 1.492e-008 kN*s/m^2;
Location1.water().density = 1.025 tonne/m^3;
Location1.water().kinematicViscosity = 1.981e-009 kN*s/m^2;
Location1.seabed().normaldirection = Vector3d(0 m, 0 m, 1 m);
Location1.seabed().seabedDelta = Scour1;
Location1.insertSoilBorder(-1.5 m);
Location1.insertSoilBorder(-3.5 m);
Location1.insertSoilBorder(-5.5 m);
Location1.insertSoilBorder(-14.5 m);
Location1.insertSoilBorder(-70 m);
Location1.insertSoilBorder(-100 m);
Location1.soil(1).soilCurves = SoilCurves1;
Location1.soil(1).soilData = SoilData1;
Location1.soil(1).soilType = Soil1;
Location1.soil(1).numberOfSublayers = 1;
Location1.soil(2).soilCurves = SoilCurves1;
Location1.soil(2).soilData = SoilData2;
Location1.soil(2).soilType = Soil1;
Location1.soil(2).numberOfSublayers = 1;

```

```

Location1.soil(3).soilCurves = SoilCurves1;
Location1.soil(3).soilData = SoilData3;
Location1.soil(3).soilType = Soil2;
Location1.soil(3).numberOfSublayers = 1;
Location1.soil(4).soilCurves = SoilCurves1;
Location1.soil(4).soilData = SoilData4;
Location1.soil(4).soilType = Soil3;
Location1.soil(4).numberOfSublayers = 3;
Location1.soil(5).soilCurves = SoilCurves1;
Location1.soil(5).soilData = SoilData5;
Location1.soil(5).soilType = Soil4;
Location1.soil(5).numberOfSublayers = 15;
Location1.soil(6).soilCurves = SoilCurves1;
Location1.soil(6).soilData = SoilData6;
Location1.soil(6).soilType = Soil5;
Location1.soil(6).numberOfSublayers = 3;
// Wave load condition
Condition1 = DeterministicTime(Location1);
Condition1.waterSurface.regularWaveSet = WaveSet1;
Condition1.populate();
Condition1.addCalmSea();
Condition1.component(1).water.current(CurrentProfile1);
Condition1.component(1).waterSurface.waveModel(Stokes5());
Condition1.component(2).water.current(CurrentProfile1);
Condition1.component(2).waterSurface.waveModel(Stokes5());
Condition1.component(3).water.current(CurrentProfile1);
Condition1.component(3).waterSurface.waveModel(Stokes5());
Condition1.component(4).water.current(CurrentProfile1);
Condition1.component(4).waterSurface.waveModel(Stokes5());
// Define analysis activity
Analysis1 = Analysis();
Analysis1.add(MeshActivity());
Analysis1.add(WaveLoadActivity(Condition1));
Analysis1.add(PileSoilAnalysis(Condition1));
Analysis1.add(LoadResultsActivity());
Analysis1.step(1).superElementType = 1;
Analysis1.step(1).superElementTypeTop = 21;
// Wave load analysis activity
Analysis1.step(2).deterministicSeastates().populate();
Analysis1.step(2).deterministicSeastates().seastate(1).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(2).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);
Analysis1.step(2).deterministicSeastates().seastate(3).dataPhase(NoStretching, 5 deg, 24, false, BothLoads, 1, 1);

```

```
Analysis1.step(2).deterministicSeastates().seastate(4).dataPhase(NoStretching,5 deg,24,false,BothLoads,1,1);
Analysis1.step(2).deterministicSeastates().seastate(5).dataPhase(NoStretching,0,1,true,NoDesignLoads,0,1);
// Create load combinations
NorMaxSh = LoadCombination(Analysis1);
NorMaxSh.addCase(Analysis1.WLC(1),1.6);
NorMaxSh.addCase(Analysis1.WLC(9),1);
NorMaxSh.addCase(Gravity,1.2);
NorMaxSh.addCase(WindN,1.6);
NorMaxMo = LoadCombination(Analysis1);
NorMaxMo.addCase(Analysis1.WLC(2),1.6);
NorMaxMo.addCase(Analysis1.WLC(9),1);
NorMaxMo.addCase(Gravity,1.2);
NorMaxMo.addCase(WindN,1.6);
EasMaxSh = LoadCombination(Analysis1);
EasMaxSh.addCase(Analysis1.WLC(3),1.6);
EasMaxSh.addCase(Analysis1.WLC(9),1);
EasMaxSh.addCase(Gravity,1.2);
EasMaxSh.addCase(WindE,1.6);
EasMaxMo = LoadCombination(Analysis1);
EasMaxMo.addCase(Analysis1.WLC(4),1.6);
EasMaxMo.addCase(Analysis1.WLC(9),1);
EasMaxMo.addCase(Gravity,1.2);
EasMaxMo.addCase(WindE,1.6);
SouMaxSh = LoadCombination(Analysis1);
SouMaxSh.addCase(Analysis1.WLC(5),1.6);
SouMaxSh.addCase(Analysis1.WLC(9),1);
SouMaxSh.addCase(Gravity,1.2);
SouMaxSh.addCase(WindS,1.6);
SouMaxMo = LoadCombination(Analysis1);
SouMaxMo.addCase(Analysis1.WLC(6),1.6);
SouMaxMo.addCase(Analysis1.WLC(9),1);
SouMaxMo.addCase(Gravity,1.2);
SouMaxMo.addCase(WindS,1.6);
WesMaxSh = LoadCombination(Analysis1);
WesMaxSh.addCase(Analysis1.WLC(7),1.6);
WesMaxSh.addCase(Analysis1.WLC(9),1);
WesMaxSh.addCase(Gravity,1.2);
WesMaxSh.addCase(WindS,1.6);
WesMaxMo = LoadCombination(Analysis1);
WesMaxMo.addCase(Analysis1.WLC(8),1.6);
WesMaxMo.addCase(Analysis1.WLC(9),1);
WesMaxMo.addCase(Gravity,1.2);
```



```
WesMaxMo.addCase(WindS,1.6);  
// Scale down basic load cases so that these do not stop Splice execution  
GenieRules.Meshing.basicLCfactor(0.01);  
*/
```